



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **CHYTRÁ DOMÁCNOST: DIGITÁLNÍ TERMOSTATICKÁ HLAVICE ŘÍZENÁ PŘES WIFI**

SMART HOME: DIGITAL THERMOSTATIC VALVE CONTROLLED VIA WIFI

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**RADIM BEDNAŘÍK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ZDENĚK MATERNA, Ph.D.**

**BRNO 2021**

## Zadání bakalářské práce



Student: **Bednařík Radim**  
Program: Informační technologie  
Název: **Chytrá domácnost: digitální termostatická hlavice řízená přes WiFi**  
**Smart Home: Digital Thermostatic Valve Controlled via WiFi**  
Kategorie: Vestavěné systémy

### Zadání:

1. Proveďte rešerši existujících řešení modifikujících levné digitální radiátorové hlavice pro jejich řízení přes WiFi, za pomoci modulů založených na ESP8266.
2. Vyberte vhodnou hlavici pro modifikaci.
3. Navrhněte úpravu, včetně napájení, způsobu řízení a odesílání naměřených dat.
4. Realizujte modifikaci hlavice a implementujte řídicí systém.
5. Ověřte dlouhodobým testem funkčnost řešení.
6. Zdrojové kódy a dokumentaci publikujte na GitHubu.
7. Vytvořte stručný plakát nebo video prezentující vaši práci, její cíle a výsledky.

### Literatura:

- Dle doporučení vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Materna Zdeněk, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

## Abstrakt

Tato bakalářská práce pojednává o návrhu a vytvoření prototypu chytré termostatické hlavice řízené přes WiFi, vzniklé modifikací levné digitální hlavice eQ-3 N. A dále o vytvoření systému, který by podporoval ovládání těchto vzniklých hlavice a umožňoval uživateli zadávat požadavky pomocí grafického rozhraní. Teoretická část práce se zabývá průzkumem aktuálního trhu nabízených termostatických hlavice, způsoby řízení vytápění pomocí řídicích algoritmů a moduly pro využití při modifikaci hlavice. Praktická část popisuje způsob modifikace hlavice a implementaci řídicího systému. Součástí práce je provedení několika testů, na ověření správného chování řízení vytápění vzniklou termostatickou hlavici.

## Abstract

This bachelor thesis deals with the design and creation of a prototype smart thermostatic valve head controlled via WiFi, created by modifying a cheap digital valve head eQ-3 N. And creating a system that would support the control of these valve heads and would allow the user to enter requirements using a graphical interface. The theoretical part deals with a survey of the current market of thermostatic valve heads offered, methods of heating control using control algorithms and modules for use in modifying the valve head. The practical part describes how to modify the valve head and the implementation of the control system. Part of the work is performing several tests to verify the correct behavior of the heating control of the created thermostatic valve head.

## Klíčová slova

Inteligentní domácnost, vestavěné systémy, ESP8266, DHT22, L298N, eQ-3 N, REST server, chytré vytápění, WiFi komunikace

## Keywords

Smart home, embedded systems, ESP8266, DHT22, L298N, eQ-3 N, REST server, smart heating, WiFi communication

## Citace

BEDNAŘÍK, Radim. *Chytrá domácnost: digitální termostatická hlavice řízená přes WiFi*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Materna, Ph.D.

# Chytrá domácnost: digitální termostatická hlavice řízená přes WiFi

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdeňka Materny, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Radim Bednařík

10. května 2021

## Poděkování

Zde bych rád poděkoval vedoucímu mé bakalářské práce Ing. Zdeňku Maternovi, Ph.D. za odborné vedení a konzultace při řešení problémů se psaním této práce, tak při praktické tvorbě.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Termostatické hlavice</b>	<b>4</b>
2.1	Mechanické . . . . .	4
2.2	Elektronické . . . . .	4
2.2.1	Digitální . . . . .	5
2.2.2	Chytré . . . . .	5
2.3	Funkcionalita . . . . .	5
2.4	Dostupná řešení . . . . .	6
2.4.1	Komerční . . . . .	6
2.4.2	Open-source a domácí . . . . .	11
<b>3</b>	<b>Algoritmy pro řízení vytápění</b>	<b>14</b>
3.1	Hysterezní řízení . . . . .	14
3.2	PID regulátor . . . . .	15
3.2.1	Varianty regulátoru . . . . .	17
3.2.2	Ladění . . . . .	17
<b>4</b>	<b>Modifikace levné digitální hlavice pro podporu chytrého řízení</b>	<b>19</b>
4.1	Hlavice pro modifikaci . . . . .	19
4.2	Modul pro řízení periférií hlavice . . . . .	19
4.2.1	Mikrokontrolér . . . . .	19
4.2.2	Pro použití v internetu věcí . . . . .	20
4.3	Řízení stejnosměrného motoru . . . . .	20
4.3.1	Princip chodu . . . . .	20
4.3.2	Ovládání . . . . .	22
4.4	Moduly . . . . .	24
4.4.1	NodeMCU . . . . .	24
4.4.2	Modul L298N . . . . .	25
4.4.3	Senzor DHT22 . . . . .	26
4.5	Modifikace . . . . .	26
<b>5</b>	<b>Návrh systému pro řízení termostatických hlavic</b>	<b>28</b>
5.1	Serverová část systému . . . . .	28
5.2	Řízení termostatické hlavice . . . . .	31
5.3	Grafického uživatelské rozhraní . . . . .	31
<b>6</b>	<b>Implementace systému pro řízení termostatických hlavic</b>	<b>32</b>

6.1	Serverová část . . . . .	32
6.2	Řízení termostatické hlavice . . . . .	34
6.3	Grafické uživatelské rozhraní . . . . .	36
<b>7</b>	<b>Testování</b>	<b>38</b>
7.1	Testování serveru . . . . .	38
7.2	Testování termostatické hlavice . . . . .	38
<b>8</b>	<b>Závěr</b>	<b>40</b>
	<b>Literatura</b>	<b>41</b>
<b>A</b>	<b>Obsah přiloženého paměťového média</b>	<b>43</b>
<b>B</b>	<b>Schéma modulu L298N</b>	<b>44</b>
<b>C</b>	<b>Zapojení modulů modifikované hlavice</b>	<b>45</b>
<b>D</b>	<b>Plakát prezentující výsledky práce</b>	<b>46</b>

# Kapitola 1

## Úvod

Při výstavbě rodinných domů, obytných komplexů nebo pracovních prostor se hledají způsoby jak co nejvíce snížit spotřebu energie. Díky internetu věcí a chytrým zařízením je v dnešních domácnostech a jiných prostorách možné využít širokou škálu chytrých zařízení, které jsou schopné dynamicky měnit podmínky prostor (systému), ve kterém se nacházejí, na základě požadavků uživatelů. V případě těchto chytrých zařízení se většinou jedná o vestavěné systémy. Vestavěný systém je ovládán zabudovaným počítačem, který byl často navrhnut pro tuto jedinou činnost. Cílem vestavěných systémů je, aby uživatel nepoznal, že přišel do kontaktu s počítačem, čehož lze často docílit aplikováním správného rozhraní mezi počítačem a uživatelem, jako je grafické rozhraní, nebo fyzická tlačítka na zařízení.

Velkou úsporu<sup>1</sup> lze dosáhnout efektivním ovládáním vytápěcího systému, jedná se o přibližně až 40% celkových výdajů na vytápění (záleží na používaných prvcích a na prostředí, ve kterém jsou používány) [2]. Pouhé snížení teploty na noc může vést k zásadním úsporám<sup>2</sup> až 20%. Existuje několik řešení, které tento problém do jisté míry řeší prostorovými termostaty, chytrými termostatickými hlavicemi, nebo chytrými bojler/kotly, které lze ovládat dálkově přes internet a dle historie používání se snaží předpovídat, kdy bude potřeba topit [7]. Výrobci těchto řešení uvádějí i dosažitelnou průměrnou úsporu díky jejich výrobkům, která se nachází mezi 10% až 40%, podle daného výrobce. Tyto řešení však často bývají dosti nákladná a uzavřená vůči dalším rozšířením.

Cílem této práce je vytvoření prototypu chytré termostatické hlavičky, modifikací již existující levné digitální termostatické hlavičky eQ-3 N, a vytvoření řídicího systému, kterým bude možné hlavičku ovládat přes síť WiFi.

Obsah práce je rozdělen do několika logických celků. Kapitola druhá definuje funkcionalitu jakou dostupné hlavičky nabízejí, popisuje rozdíl mezi mechanickými, digitálními a chytrými hlavicemi, a věnuje se průzkumu aktuálního trhu a existujícím řešením. V kapitole třetí jsou popsány dva nejrozšířenější algoritmy pro řízení lokální teploty, jejich výhody/nevýhody a jakým způsobem fungují. Modifikace hlavičky eQ-3 N je popsána v kapitole čtvrté společně s prvky, které jsou na modifikaci využity a technologiemi, na kterých jsou tyto prvky založené. Návrh řídicího systému je obsažen v kapitole páté, ve které je popsán návrh serveru a jeho rozhraní, a řídicí program hlavičky. Kapitola šestá se zabývá implementací systému, jehož návrh je popsán v předešlé kapitole. A v poslední části je popis testování systému.

---

<sup>1</sup>Příklad článku popisujícího výhody a dosažitelné úspory chytrého vytápění: <https://www.self-build.co.uk/can-smart-heating-save-you-money/>

<sup>2</sup><https://vytapieni.tzb-info.cz/vytapime-elektrinou/21604-vliv-prerusovaneho-vytapeni-na-spotrebu-energie-pri-vytapeni-administrativni-budovy>

## Kapitola 2

# Termostatické hlavice

Tato kapitola představuje a popisuje různé mechanické i elektronické termostatické hlavice, jakou funkcionalitu lze běžně u takových hlavice nalézt, jaké typy existují a průzkum trhu termostatických hlavice, jejich vlastností a nabízených funkcí.

### 2.1 Mechanické

Mechanické termostatické hlavice nabízejí relativně spolehlivé jednoduché řízení vytápění. Ačkoliv jsou mechanické, tak se nejedná pouze o prvek, který ovládá radiátorový ventil staticky.

Typická mechanická hlavice se skládá z obalu, na kterém je vyobrazena stupnice, a z rozpínavé hmoty uvnitř (může se jednat například o líh, či o typ vosku), která mění svůj objem v závislosti na teplotě a tím způsobuje stlačení/povolení ventilu a reguluje tak vstup teplé vody do radiátoru.

Stupnice na mechanické hlavici udává za jaké okolní teploty se má stále pouštět do radiátoru teplá voda. Každá hodnota stupnice tedy udává maximální teplotu, na kterou se má vytápět (viz tabulka 2.1). Při nastavení stupnice se ventil stlačí na tolik, aby při dosažení maximální teploty, byla rozpínavá látka v takovém stavu, že stlačí ventil až do uzavřeného stavu a teplá voda přes něj nebude proudit.

Stupnice	Hlavice Danfoss
0	7°C
1	13°C
2	17°C
3	20°C
4	23°C
5	26°C

Tabulka 2.1: Maximální teplota při hodnotách stupnice mechanické termostatické hlavice

### 2.2 Elektronické

Elektronické termostatické hlavice existují v několika typech, které se liší typem přístupu a rozsahem nabízených funkcionalit.



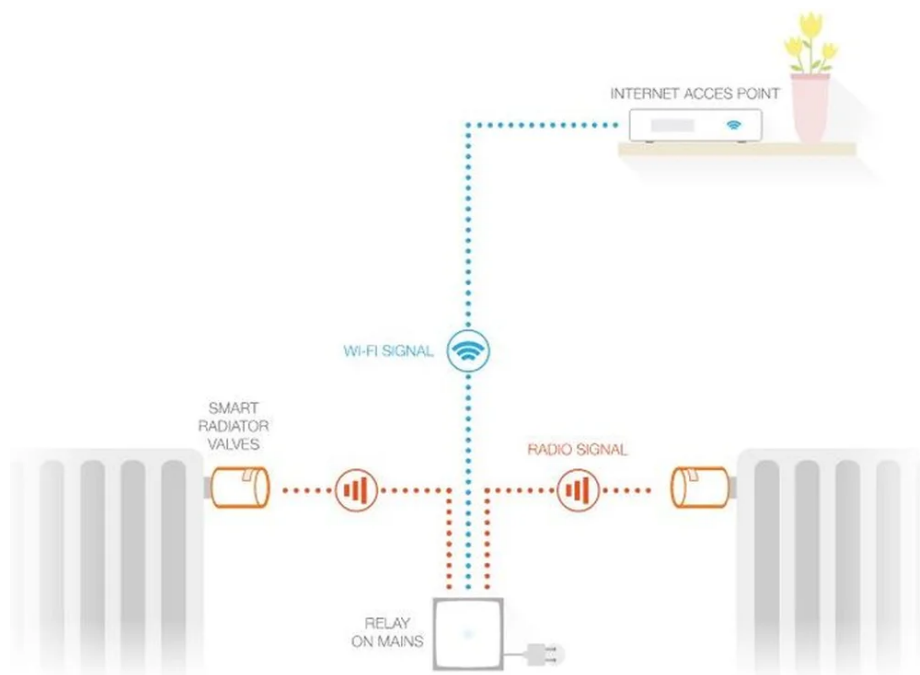
### 2.2.1 Digitální

Digitální hlavici lze chápat jako lokální řídicí prvek topného systému. Přístup k digitální hlavici je realizován fyzicky, kdy je potřeba své požadavky zadat osobně přes ovládací rozhraní na hlavici.

### 2.2.2 Chytré

Chytré termostatické hlavice jsou takové, ke kterým lze přistoupovat vzdáleně, ať už přes internet, či bluetooth. Pro přístup přes internet je však téměř vždy potřeba internetové brány/relé pro komunikaci, se kterou budou hlavice komunikovat a která je bude propojovat z lokální sítě přes router do internetu.

Výhodou chytrých systémů (příklad takového systému na obrázku 2.1) je, že jsou rozšiřitelné, jak o zařízení stejného typu, tak rozdílného. Také je možné pro jejich chod použít data z internetu, pokud to je vhodné u daného zařízení, v případě termostatických hlavice se může jednat o aktuální počasí a teplotu v místě, kde se hlavice nachází.



Obrázek 2.1: Příklad komunikace systému Netatmo<sup>1</sup>

## 2.3 Funkcionalita

Elektronické termostatické hlavice z pravidla nabízejí časové plány vytápění a lepší vytápění díky řídicím algoritmům, kterým lze dosáhnout jak zvýšení komfortu pro uživatele, tak k relativně velké úspoře spotřebované energie. Často také nabízejí různé režimy pro většinu

<sup>1</sup>Převzato z <https://netatmostatic.blob.core.windows.net/static/pro/NRV/Training-Book-NRV-EN.pdf>

běžných potřeb, jako je krátká nepřítomnost v domě, kdy dojde k poklesu od standardní teploty, nebo režim přítomnosti v místnosti, kdy dochází k mírnému zvýšení od standardní teploty.

**Individuální program** Individuální program definuje časové úseky, na kterých si lze určit rozdílné teploty. Nejčastěji se jedná o hodiny v rozdílných dnech týdne, ale také se někdy jedná o několik úseků dne v týdnu.

**Dynamická regulace teploty** Při regulaci teploty jsou brány v potaz některé vnější aspekty vytápěného prostředí, díky čemuž se dosahuje přesnějšího a úspornějšího topení, mezi něž může například patřit venkovní teplota, počasí, či přítomnost osoby v místnosti.

**Detekce otevřeného okna** Detekce otevřeného okna je funkce elektronických hlavice, která detekuje větrání v místnosti buď na základě snímače kontaktu v okně, nebo na základě rychlého výkyvu teploty a vlhkosti v místnosti, a následně zavře ventil topení, aby nedocházelo ke zbytečnému vytápění. Po detekci otevřeného okna se topení vrátí do standardního režimu buď po daném časovém intervalu, nebo po sepnutí kontaktu snímače v okně, a nebo po ustálení teploty a vlhkosti na vnitřní stav místnosti.

**Ochrana ventilu** Režim proti zamrznutí chrání ventil při nízkých teplotách tím, že udržuje topení převážně nad teplotou přibližně 7°C. Při delším nepoužívání topení, jako je v létě, kdy není potřeba vytápět, se ventil úplně otevře a pootočí v daných časových intervalech, aby nedošlo k poškození ventilu a také hlavice.

**Režimy** Většina elektronických hlavice obsahuje několik režimů pro usnadnění regulace mezi často požadovanými teplotami. Hlavice mává normální režim, kdy se udržuje zadaná teplota, dále obsahuje režim ekonomický, který žádanou teplotu sníží o několik stupňů, z důvodu nepřítomnosti osoby v místnosti či domě, a dále režim zvýšené teploty, pro potřebu rychlého vytopení místnosti, nebo z důvodu požadavků jiné osoby. Některé hlavice obsahují i další režimy navíc, ale tyto základní má většina hlavice společných.

## 2.4 Dostupná řešení

Na trhu se nachází velké množství různých produktů, lišící se nabízenými funkcemi, způsobem ovládání, cenou a dalšími aspekty. Existují však také open-source řešení a takzvané DIY (Do It Yourself) domácí řešení.

### 2.4.1 Komerční

Existuje veliký počet existujících komerčních řešení termostatických hlavice. V této části jsou popsány některé z nich.

#### eQ-3 N

Jedná se o digitální termostatickou hlavici od firmy Eqlva. Ovládat tuto hlavici lze pouze přes fyzické rozhraní na hlavici. Hlavice nabízí celou škálu funkcí, mezi hlavní patří časový

program, ve kterém lze v každém rozdílném dni v týdnu měnit teplotu mezi dvěma hodnotami až sedmkrát. Mezi další funkce hlavičky patří ochrana před mrazem, režim dovolené a detekce otevřeného okna.



Obrázek 2.2: Termostatická hlavička eQ-3 N<sup>2</sup>

<b>Typ</b>	digitální
<b>Přístup</b>	fyzicky
<b>Funkce</b>	týdenní program
<b>Napájení</b>	baterie
<b>Cena</b>	360kč

Tabulka 2.2: Parametry termostatické hlavičky eQ-3 N

**Výhody/Nevýhody** Výhodou této hlavičky je rozhodně cena, na trhu se nenachází hlavičky s podobně širokou funkcionalitou za tak nízkou cenu. Avšak jedná se o digitální hlavičku, takže není možnost vzdáleného přístupu a ovládání.

### Danfoss Eco BT

Chytrá hlavička Danfoss Eco BT, jak již jméno napovídá, používá pro vzdálený přístup komunikaci přes bluetooth. Podporuje funkci otevřeného okna pro zamezení zbytečného vytápění během větrání, režim dovolené který sníží teplotu, dokud nebude zrušen. Hlavička dále umožňuje ovládání vytápění pomocí pokročilého algoritmu PID regulátoru.

<sup>2</sup>Převzato z <https://www.nej-ceny.cz/697329/programovatelná-termostatická-hlavička-eq-3-n.html>



Obrázek 2.3: Termostatická hlavice Danfoss Eco BT<sup>3</sup>

<b>Typ</b>	chytrá
<b>Přístup</b>	fyzicky, mobilní aplikace
<b>Bezdrátová komunikace</b>	bluetooth
<b>Funkce</b>	PID regulátor, detekce otevřeného okna
<b>Napájení</b>	baterie
<b>Cena</b>	900kč

Tabulka 2.3: Parametry termostatické hlavice Danfoss Eco BT

**Výhody/Nevýhody** Výhodou je cena, která je jedna z nižších mezi hlavicemi, které je možné ovládat vzdáleně přes telefon, ale ovládání je pouze přes bluetooth s výrobcem specifikovaným dosahem deset metrů, což ve větších domech někdy může být problém, a tím pádem hlavici nelze ovládat přes internet.

### Honeywell Evohome HR92EE

Jedná se o chytrou hlavici, kterou lze ovládat přes mobilní aplikaci, pomocí komunikace přes rádiové vlny o frekvenci 868 až 870 MHz, lze s ní komunikovat také přes internet, ale pro to je potřeba vlastnit také řídicí jednotku Honeywell Evohome<sup>4</sup>, která bude připojena k internetu a se kterou bude hlavice spárována. Hlavice Honeywell Evohome H92EE nabízí týdenní program rozdělený na hodiny pro detailnější nastavení. Hlavice dále nabízí řadu režimů, jako režim nepřítomnosti doma či komfortní režim. Nabízí detekci otevřeného okna, ale pouze přes kontaktní snímač Honeywell.

<sup>3</sup>Převzato z <https://www.mall.cz/regulace-vytapeni/danfoss-eco-bluetooth-3x-inteligentni-radiatorova-termostaticka-hlavice-bila>

<sup>4</sup><https://getconnected.honeywellhome.com/cz/evohome-wi-fi.html>



Obrázek 2.4: Termostatická hlavice Honeywell Evohome HR92EE<sup>5</sup>

<b>Typ</b>	chytrá
<b>Přístup</b>	fyzicky, mobilní aplikace, řídicí jednotka
<b>Bezdrátová komunikace</b>	rádiové vlny 868-870 MHz
<b>Funkce</b>	týdenní program
<b>Napájení</b>	baterie
<b>Cena</b>	1500kč

Tabulka 2.4: Parametry termostatické hlavice Honeywell Evohome HR92EE

**Výhody/Nevýhody** Honeywell Evohome je chytrá hlavice, její ovládání je tedy pohodlné a praktické. Mezi chytrými hlavici patří k těm levnějším. Nevýhodou je potřeba kontaktního snímače pro použití detekce otevřeného okna, který stojí okolo 500 korun.

#### Netatmo chytrá radiátorová hlavice

Už z jména je zřejmé, že se jedná o chytrou termostatickou hlavici. Hlavice nabízí čtyři režimy nastavení teploty, režim noc, ekonomický, komfort a kombinaci ekonomický+komfort, tyto režimy je možné zakomponovat do individuálního plánu. Dále hlavice nabízí detekci otevřeného okna a jak hysterezní řízení teploty, tak řízení pomocí PID regulátoru. Hlavici lze ovládat manuálně, avšak ne všechnu její funkcionalitu, dále přes mobilní aplikaci, webové rozhraní a také je kompatibilní s aplikacemi chytré domácnosti jako Google Home nebo Apple HomeKit a s dalšími podobnými.

<sup>5</sup>Převzato z <https://www.czc.cz/honeywell-evohome-hr92ee-termostaticka-hlavice/192379/produkt>



Obrázek 2.5: Chytrá termostatická hlavice Netatmo<sup>6</sup>

<b>Typ</b>	chytrá
<b>Přístup</b>	fyzicky, mobilní aplikace, web
<b>Bezdrátová komunikace</b>	WiFi 2.4 GHz
<b>Funkce</b>	týdenní program, PID regulátor
<b>Napájení</b>	baterie
<b>Cena</b>	2000kč

Tabulka 2.5: Parametry chytré termostatické hlavice Netatmo

**Výhody/Nevýhody** Hlavice má široký záběr funkcionalit a podporu dvou rozdílných metod řízení topení. Nevýhodou je, že výrobce píše, že pro správné fungování je potřeba vlastnit řídicí jednotku a být připojen k internetu. Jedná se o jednu z dražších hlavice na trhu, ale také o jednu z kvalitnějších.

### Tado chytrá termostatická hlavice

Chytrá termostatická hlavice Tado nabízí nastavení individuálního plánu, pro řízení teploty podle preferencí uživatele. Nabízí funkci otevřeného okna a funkci geofencing, při odchodu z domu uživatele upozorní, zda nechce snížit topení, mezi těmito standardními funkcemi lze u této hlavice najít kontrolu a doporučení ohledně kvality vzduchu v domě, a při řízení topení bere v potaz místní předpověď počasí, aby se například netopilo zbytečně navíc, když má být slunečno.

<sup>6</sup>Převzato z <https://www.netatmo.com/cs-cz/energy/additional-valve>



Obrázek 2.6: Chytrá termostatická hlavice Tado<sup>7</sup>

<b>Typ</b>	chytrá
<b>Přístup</b>	fyzicky, mobilní aplikace
<b>Bezdrátová komunikace</b>	rádiové vlny 868-870 MHz, WiFi 2.4 GHz
<b>Funkce</b>	týdenní program
<b>Napájení</b>	baterie
<b>Cena</b>	3200kč(i s internetovou bránou)

Tabulka 2.6: Parametry chytré termostatické hlavice Tado

**Výhody/Nevýhody** Hlavice oproti ostatním nabízí některé unikátní funkce jako je například geofencing. Hlavice podporuje jak hysterezní řízení, tak i řízení pomocí PID regulátoru. Jedná se však o hlavici ve vyšší cenové relaci, a nelze pořídit pouze hlavici, je nutné koupit v balení i s internetovou bránou.

## 2.4.2 Open-source a domácí

Mimo komerčních řešení existují také open-source řešení a podomácku vytvořená řešení a v této části jsou některá z nich představena.

### OpenTRV

OpenTRV<sup>8</sup> je open-source hardwarové i softwarové řešení zónového vytápění, pomocí bezdrátových termostatických hlavic (příklad provedení hlavice viz 2.7). Řešení je mířené na domácnosti s vlastním zdrojem tepla. Cílem OpenTRV projektu je nabízet nenákladnou možnost, jak účinně vytápět jednotlivé prostory v domě či bytě, a tedy i snížit spotřebovanou energii a emise uhlíku.

Z počátku řešení obsahovalo zařízení, které komunikovalo mezi hlavicí třetí strany (například Conrad FHT8V) a kotlem. Zařízení obsahovalo teplotní senzor, podle jehož hodnot ovládalo hlavici a kotel. Aktuální řešení je realizováno pomocí hlavice s vestavěným systémem, které komunikuje se zařízením ovládající kotel, toto řešení obsahuje senzor teploty a

<sup>7</sup>Převzato z <https://www.scan.co.uk/products/tado-smart-radiator-thermostat-vertical-wireless-battery-powered>

<sup>8</sup><https://github.com/opentrv>

pohybu, podle kterého se snaží dostatečně vytápět pouze, když se v místnosti nachází osoba. Do budoucna si dává projekt za úkol vytvořit chytrou termostatickou hlavici, kterou bude možné ovládat bezdrátově přes internet.



Obrázek 2.7: Termostatická hlavice OpenTRV<sup>9</sup>

**Výhody/Nevýhody** To že je řešení open-source je rozhodně výhodou, protože to zaručuje nízkou cenu na vytvoření, ale je potřeba si zařízení vytvořit sám. Řešení není interaktivní vůči uživateli, tudíž jej není možné ovládat přímo.

### DIY chytrá termostatická hlavice

Existuje mnoho domácích řešení termostatických hlavic, jeden příklad je chytrá hlavice<sup>10</sup> s podporou softwaru Home Assistant<sup>11</sup> (provedení viz 2.8. Řešení nabízí nastavení jednotlivých teplot v dané časové úseky, bezdrátové ovládání pomocí Home Assistant, ovládání vytápění se pouze zapíná/vypíná při překročení požadované teploty ze shora/zdola.

<sup>9</sup>Převzato z <http://www.earth.org.uk/open-source-programmable-thermostatic-radiator-valve.html>

<sup>10</sup><https://www.instructables.com/Smart-Radiator-Valve-With-Home-Assistant/>

<sup>11</sup><https://www.home-assistant.io>





Obrázek 2.8: Chytrá DIY termostatická hlavice<sup>12</sup>

**Výhody/Nevýhody** Jedná se o chytrou hlavici, která je ovládána pomocí bezplatného softwaru, není obtížná na vytvoření, protože se skládá z již hotových modulů, které je potřeba pouze zapojit mezi sebou. Obsahuje pouze jednoduchou funkcionalitu a pro vytvoření schránky pro elektroniku je potřeba 3d tiskárny.

---

<sup>12</sup>Převzato z: <https://www.instructables.com/Smart-Radiator-Valve-With-Home-Assistant/>

## Kapitola 3

# Algoritmy pro řízení vytápění

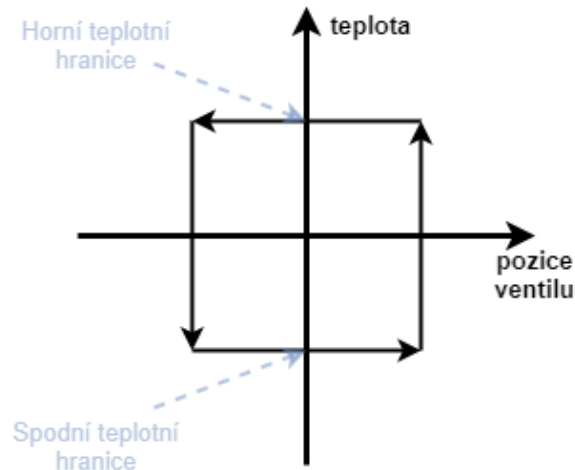
V této kapitole si představíme několik algoritmů, kterými lze řešit regulaci vytápění v domácnosti s omezenou znalostí řízeného systému (velikost místnosti, natočení oken atd.). Hlavní aspekty řízení vytápění jsou intervaly ve kterých se mění stav řízení (změna pozice ventilu v topení) a průměrná přesnost dosažení požadované hodnoty, tyto aspekty jsou si do určité míry protichůdné, nelze dosáhnout obou výborně zároveň, proto existuje několik algoritmů, které dávají přednost přesnosti, nebo delším intervalům mezi změnami [12].

### 3.1 Hysterezní řízení

Hysterezní řízení je označení pro řízení, jehož výstupní hodnota nezáleží pouze na aktuálním stavu řízeného systému, ale také na předchozím stavu (viz obrázek 3.1). Používá se pro řízení, aby nedocházelo k malým změnám řízení a často, ale k větším a řízení se bude měnit s většími odstupy.

V řízení vytápění to znamená, že se specifikují dvě teplotní hranice, jedna pro zapnutí vytápění a druhá pro vypnutí vytápění. Když teplota klesne pod spodní hranici, tak se topení zapne a začne vytápět, když teplota překročí horní hranici a topení se vypne.

Pokud se horní hranice a spodní hranice rovnají, tak se řízení říká zapnuto/vypnuto, je to takové řízení, u kterého existuje pouze jedna hranice, který odděluje zda se vytápí či nikoliv. Nulová hranice se příliš nepoužívá, protože dochází k častým změnám a tedy drasticky snižuje životnost mechanických prvků.



Obrázek 3.1: Hystereze vytápění

## 3.2 PID regulátor

PID regulátor je řídicí algoritmus, fungující na základě tří složek, **P**roporcionální, **I**ntegrační a **D**erivační [1]. Algoritmus v pravidelných intervalech počítá regulační odchylku, která je rozdílem mezi žádanou hodnotou a získaným výstupem měřené veličiny, na jejímž základě se společně s proporcionálním, integračním a derivačním koeficientem vypočítá vstup, který bude použit pro nastavení v daném systému.

**Proporcionální složka** Proporcionální složka produkuje výstupní hodnotu, která je přímo úměrná velikosti regulační odchylky a je dána rovnicí:

$$P = K_p * e(t) \quad (3.1)$$

Pro dosažení požadovaného výsledku se mění proporcionální koeficient značený  $K_p$ . Jeho změna má velký dopad na výslednou hodnotu. Vyšší proporcionální koeficient zajišťuje rychlejší nástup žádané hodnoty, ale také způsobuje její přestřel a déle trvá přechod k ustálené hodnotě.

S proporcionální složkou se používá proporcionální pás, který ohraničuje žádanou hodnotu z obou stran. Pokud je regulační odchylka menší než hodnota proporcionálního pásu, proporcionální složka bere, že byla dosažena žádaná hodnota, toto v reálném použití způsobuje, že se hodnota drží na hraně proporcionálního pásu, tedy nedosahuje žádané hodnoty a pouze se drží pod žádanou hodnotou, tomuto problému se říká chyba ustáleného stavu.

**Integrační složka** Výstupní hodnota integrační složky je přímo úměrná integrálu regulační odchylky v časovém intervalu  $(0, t)$ . Integrační složka je dána rovnicí

$$I = K_i * \int_0^t e(t) dt \quad (3.2)$$

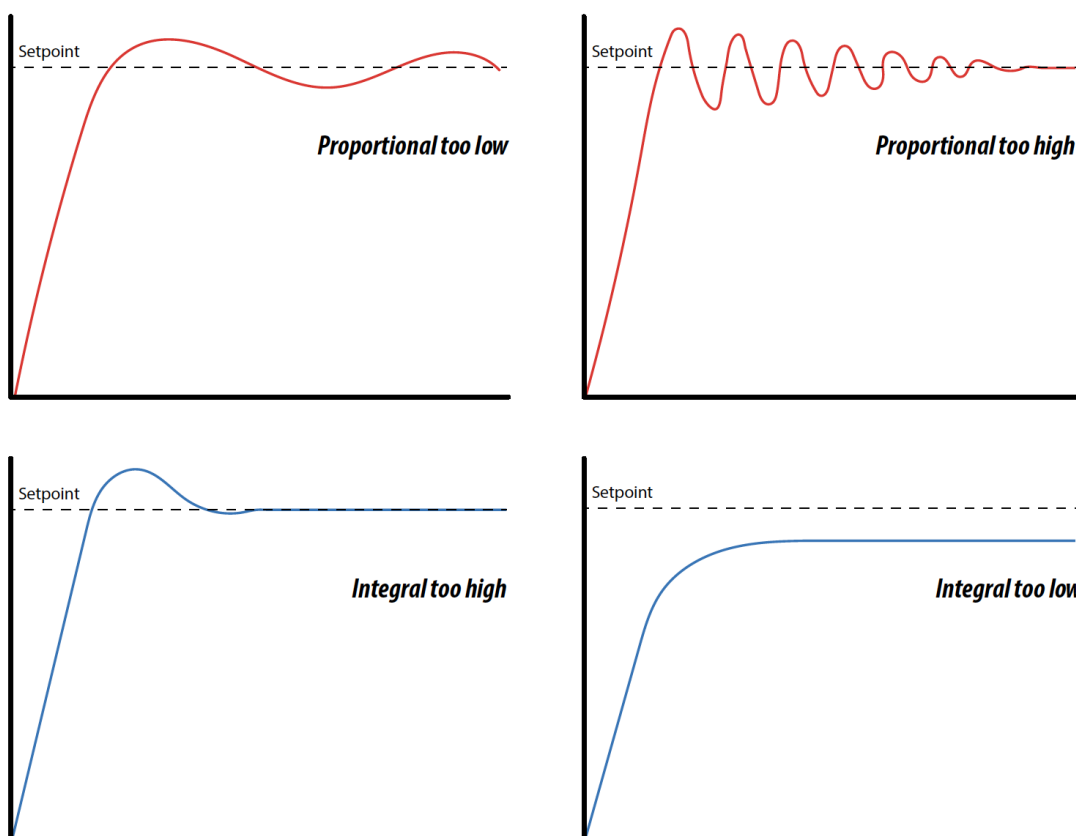
Integrační složka zrychluje náběh na žádanou hodnotu a řeší chybu ustáleného stavu, tím že se zvětšuje v době, kdy se nachází pod žádanou hodnotou, tedy i na hraně proporcionálního pásu. Ale stálým zvětšováním integrační složky ve stavu náběhu, kdy se přičítá

regulační chyba, dochází k přestřelu žádané hodnoty, toto se často řeší omezením hodnoty integrační složky ze shora, aby hodnota nerostla více než je potřeba.

**Derivační složka** Výstupní hodnota derivační složky je definována aktuálním růstem/-klesáním sledované veličiny, tedy její derivací. Derivační složka je dána rovnicí

$$D = -K_d * \frac{de(t)}{dt} \quad (3.3)$$

Derivační složka slouží k odhadu, kde se bude sledovaná veličina nacházet v budoucnosti, jak daleko do budoucnosti se bude dívat lze upravit derivačním koeficientem  $K_p$ . Cílem této složky je zamezit přestřelu při náběhu na žádanou hodnotu a rychlejšímu nastolení správného ustáleného stavu. Nastavením většího derivačního koeficientu než je potřeba dochází k velkým chybám, které jsou způsobeny příliš velkým předpovídáním a i v téměř ustáleném stavu vidí velikou chybu.



Obrázek 3.2: Příklad grafů řízených PID regulátorem<sup>1</sup>

<sup>1</sup>Převzato z <https://clippard.com/cms/wiki/practical-pid-process-dynamics-proportional-pressure-controllers>

### 3.2.1 Varianty regulátoru

Od PID regulátoru se odvozuje několik dalších regulátorů, které buď přímo využívají některé již existující složky, nebo staví další složky nad již existujícími.

**P regulátor** P regulátor využívá pouze proporcionální složku, což přináší výhody i nevýhody. P regulátor je jednoduchý na naladění, protože obsahuje pouze jednu složku, tudíž je nutné sledovat a měnit pouze jednu hodnotu. Je vhodný do systémů, kde není nutné nastolit stabilní stav rychle a kde není problémem náběhový přestřel, nebo delší náběh na žádanou hodnotu.

**PI regulátor** PI regulátor ze standardního PID regulátoru vynechává derivační složku. Toto rozhodně přináší výhodu jednoduššího naladění, protože derivační složku je nejobtížnější zvolit tak, aby ji nebylo moc nebo málo. Přidání integrační složky k P regulátoru řízení získá rychlejší ustálení stavu. Ale při požadavku na rychlý náběh na ustálený stav, zůstává problém náběhového přestřelu, ale i přes to se jedná v provozu nejpoužívanější variantu PID regulátoru.

**PID regulátor** Využívá všechny složky, což vede k přesnému řízení systému ve všech směrech. U tohoto regulátoru je jedinou větší nevýhodou jeho ladění, které bývá velice obtížné.

### 3.2.2 Ladění

Ladění PID regulátoru je velice důležité k dosažení požadovaných výsledků, ale zároveň velmi obtížné. Proto se v reálném provozu nejčastěji používá pouze PI regulátor, který je díky absenci derivační složky jednodušší na nastavení, a také kvůli tomu, že mnohdy se nachází v systému nějaký šum ve sledované veličině, díky kterému by ani regulátor s derivační složkou nemohl správně fungovat, PI regulátor je také mnohdy dostačující, protože v mnoha reálných systémech se ani nenachází, nebo není potřeba brát v potaz vnější jev, který by řízení systému nečekaně dostatečně ovlivňoval. Často se také používá pouze P regulátor, protože je v některých systémech dostačující a není pro jeho vyladění potřeba žádná metoda, neboť se dá jednoduše nastavit manuálně.

Jednou z populárních metod ladění je Zigler-Nicholsova metoda [3], která funguje následovně

1. Z počátku se nastaví integrační a derivační konstanta na nulu a proporcionální konstanta na malou hodnotu.
2. Postupně se zvětšuje proporcionální hodnota, dokud systém nedosáhne postupné periodické oscilace až ke stabilnímu stavu.
3. Hodnotu proporcionální konstanty si označíme jako  $K_u$  a frekvenci periody jako  $f_u$ .
4. Hodnoty všech koeficientů nastavíme dle následující tabulky.

Typ regulátoru	$K_p$	$K_i$	$K_d$
P	$0.5K_u$	$\times$	$\times$
PI	$0.45K_u$	$1.2f_u$	$\times$
PID	$0.6K_u$	$2.0f_u$	$0.125/f_u$

Tabulka 3.1: Tabulka pro nastavení PID regulátoru Zigler-Nicholsovou metodou

Existují však systémy, u kterých takto nastavený regulátor nemusí fungovat správně.

## Kapitola 4

# Modifikace levné digitální hlavice pro podporu chytrého řízení

Kapitola je zaměřena na modifikaci vybrané digitální hlavice, a přiblížení technologií a modulů využívajících tyto technologie, potřebných k chytrému provozu modifikované termostatické hlavice. Mezi tři hlavní komponenty patří řídicí prvek, kterým je v internetu věcí nejčastěji mikrokontrolér, elektrický motor pro pohyb s ventilem topení a komponenta pro měření teploty a vlhkosti v okolním prostředí.

### 4.1 Hlavice pro modifikaci

Jako hlavici pro modifikaci jsem zvolil eQ-3 N. Hlavním důvodem byla cena, která je aktuálně mezi nejnižšími na trhu.

### 4.2 Modul pro řízení periférií hlavice

Hlavní částí modifikované hlavice je mikrokontrolér, který bude ovládat všechny ostatní prvky hlavice a bude řídit její chod, tedy teplotu pomocí regulace ventilu na topení.

#### 4.2.1 Mikrokontrolér

Mikrokontrolér je kompaktní integrovaný obvod obsahující kompletní počítač [6]. Je navržen tak, aby vykonával specifické funkce ve vestavěných systémech. Mikrokontrolér se většinou skládá ze tří základních částí, centrální procesorové jednotky, volatilní a nevolatilní paměti, a vstupně-výstupních periférií.

Procesorová jednotka vykonává aritmetické operace, manipuluje s daty a vykonává další logické operace na základě zadaných instrukcí.

Paměť je rozdělena na nevolatilní, která obsahuje program, který bude mikrokontrolér vykonávat, jeden mikrokontrolér má často jeden program po celou dobu svého chodu, a na volatilní sloužící k ukládání dat potřebných v průběhu programu, jako jsou například proměnné programu. Paměť je často pouze částí procesorové jednotky.

Periférie slouží ke komunikaci s okolními zařízeními a k jejich ovládání. Mezi klasické periférie patří datové převodníky (analogový signál na digitální a naopak), univerzální vstupně-výstupní periférie pro ovládání připojených zařízení a periférie sériové komunikace jako UART, SPI nebo I2C.

### 4.2.2 Pro použití v internetu věcí

Aby byl mikrokontrolér použitelný v IoT, tak musí obsahovat modul, který podporuje nějakou formu bezdrátové komunikace. Nejpopulárnější bezdrátovou komunikací je díky své jednoduché použitelnosti WiFi, tento typ komunikace také využívá nejpopulárnější mikrokontrolér pro tvorbu chytrých zařízení ESP8266 od firmy Espressif System.

Mezi dva známé moduly založené na ESP8266 patří NodeMCU a WEMOS D1, které si jsou nabízenou funkcionalitou velmi podobné, největším rozdílem je, že NodeMCU nabízí více pinů pro obecné použití.

## 4.3 Řízení stejnosměrného motoru

V této podkapitole jsou představeny typy stejnosměrných motorů a způsoby jak tyto motory spolehlivě řídit.

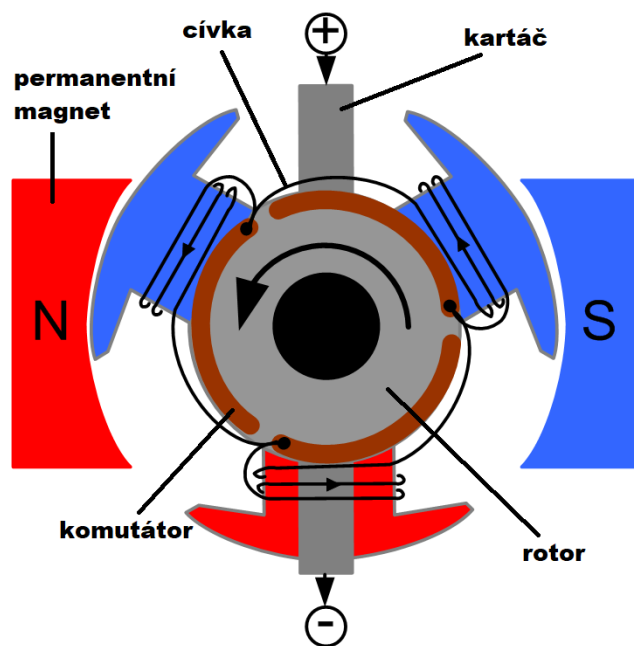
### 4.3.1 Princip chodu

Existují dva základní typy stejnosměrných motorů, kartáčový a bezkartáčový, oba pracují na stejném principu, ale rozdílným způsobem [4].

#### Kartáčový stejnosměrný motor

Kartáčový motor se skládá ze dvou permanentních magnetů (statoru) na opačných stranách motoru, uprostřed se nachází rotor s cívkami, které jsou připojeny ke komutátoru, přes který je jimi veden proud, na komutátor je veden proud skrze kartáče (viz obrázek 4.1). Jakmile je do cívky přiveden proud, tak cívka začne vytvářet magnetické pole, jelikož se rotor s cívkou nachází v magnetickém poli statoru, tak se rotor začne otáčet, tak aby se cívky dostaly do polohy nejmenšího odporu vůči statoru. Tím že se rotor otáčí, se otáčí i komutátor, který přepne polaritu napájení na cívce, čímž změni i polaritu magnetického pole a začne být přitahován k protilehlému magnetu než doposud. Tímto průběžným měněním polarity se u bezkartáčového stejnosměrného motoru dosahuje plynulé rotace. Pro dosažení plynulého pohybu je také potřeba mít nejméně tři cívky, aby nedocházelo ke stavu, kdy se cívky nacházejí přesně mezi magnety.





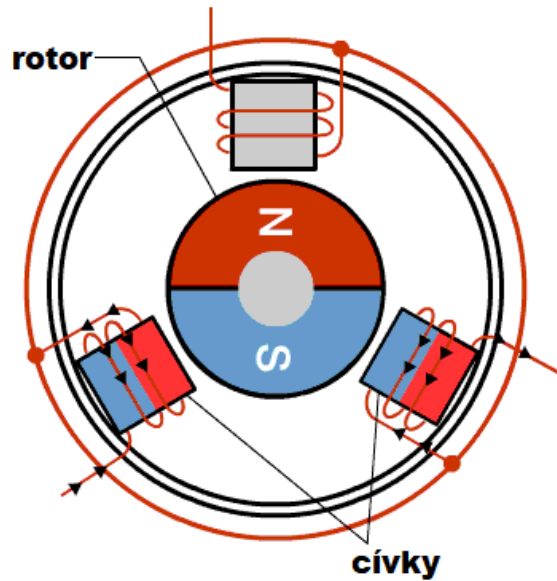
Obrázek 4.1: Struktura kartáčového stejnosměrného motoru<sup>1</sup>

### Bezkartáčový stejnosměrný motor

Bezkartáčový motor se skládá z několika cívek, které jsou pevně usazené, a permanentního magnetu. Jsou dva základní typy, kde u jednoho se cívky nacházejí uprostřed motoru a magnety okolo nich (viz obrázek 4.2), a u druhého se magnety nacházejí uprostřed a cívky okolo nich, vždy je ale pohyblivou částí magnet a ne cívky jak tomu bylo u kartáčové varianty motoru.

Cívkami prochází proud tak, aby postupně přitahovaly magnety na rotoru, čímž se dosáhne plynulého pohybu.

<sup>1</sup>Převzato z <https://www.seeedstudio.com/blog/2019/10/08/1298-all-about-1298-motor-driver/>



Obrázek 4.2: Struktura bezkartáčového stejnosměrného motoru s magnety uprostřed<sup>2</sup>

Výhodou této verze je, že mezi žádnými částmi motoru nedochází ke kontaktu, jak tomu bylo u kartáčového motoru mezi kartáčem a komutátorem, a tedy nedochází k opotřebení způsobeným fyzickým kontaktem. Nevýhodou je však obtížnější ovládání a potřeba senzoru Hallova jevu, pro určení pozice rotoru v motoru.

### 4.3.2 Ovládání

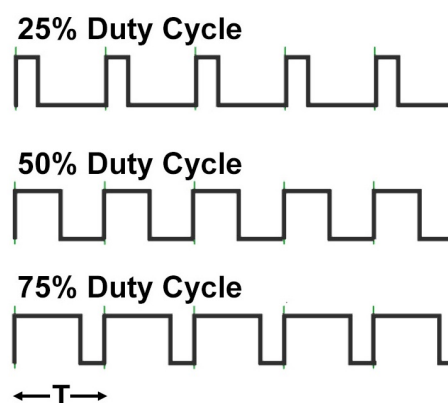
Pro plnou kontrolu nad motorem je potřeba ovládat rychlost a směr otáčení motoru, pro tyto účely se nejčastěji používá pulzně šířková modulace pro regulaci rychlosti, a H-můstek pro ovládání směru otáček motoru.

#### Pulzně šířková modulace

Pulzně šířková modulace je metoda pro regulaci průměrné elektrické energie, dodávanou dvoustavovým elektrickým signálem.

Digitální signál dokáže být pouze ve dvou stavech, buď je ve stavu vysokého napětí, nebo nízkého (nulového). Pulzně šířková modulace v rychlých cyklech zapíná a vypíná napětí signálu (viz 4.3), čímž se dosáhne napodobení analogového signálu, protože výsledné efektivní napětí je zprůměrování všech hodnot, ve kterých se napětí nacházelo v jednom cyklu. Výsledný signál po použití modulace je periodicky se opakující schodový signál.

<sup>2</sup>Převzato z <https://automatizace.hw.cz/ec-technologie-nebo-frekvencni-menici.html>

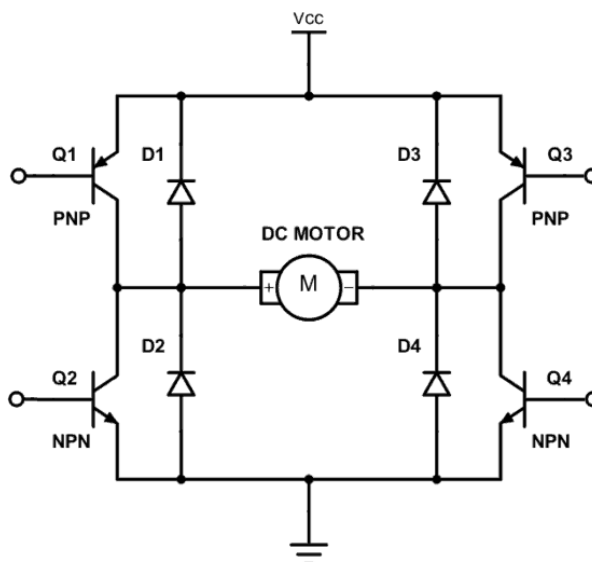


Obrázek 4.3: Příklad pulzně šířkově modulovaných signálů<sup>3</sup>

## H-můstek

H-můstek je jednoduchý elektrický obvod používaný k řízení směru pohybu stejnosměrných kartáčových motorů [9].

Základními prvky H-můstku jsou čtyři tranzistory, které mění cestu elektrického proudu tak, aby proud procházel motorem v žádaném směru. Pokud se nastaví tranzistor Q1 a Q4, dle obrázku 4.4, do propustného stavu, tak se bude motor točit jedním směrem, a pokud budou v propustném stavu tranzistory Q3 a Q2, tak se bude točit opačným směrem.



Obrázek 4.4: H-můstek<sup>4</sup>

<sup>3</sup>Převzato z <https://nightsea.com/articles/pwm-led-dimming/>

<sup>4</sup>Převzato z <https://www.build-electronic-circuits.com/h-bridge/>

Při používání H-můstku je nutné dávat pozor na některé stavy. Pokud se otevře tranzistor Q1 a Q2, tak vznikne cesta mezi napájecím zdrojem a zemí, s velmi nízkým odporem a může dojít ke zkratu, z tohoto důvodu se v praxi neovládají jednotlivé tranzistory zvlášť, ale vždy jeden pár na jedné straně společně. H-můstek je zpravidla ovládán třemi signály, jeden je povolovací a dva jsou pro ovládání směru proudu. Na jedné straně se před každým tranzistorem nachází logický člen AND, do kterého vede povolovací signál a jeden ovládací signál, respektive negace ovládacího signálu, a zrcadlově to je stejně na druhé straně H-můstku, jen je zde veden druhý ovládací signál. Toto tedy neumožňuje, aby na jedné straně byly aktivní oba dva tranzistory.

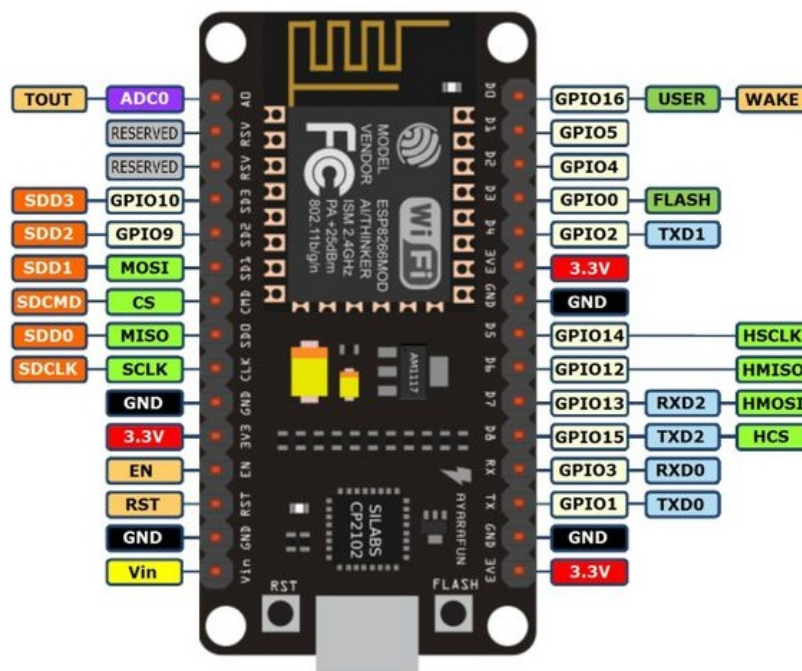
Motor při chodu indukci vytváří elektrickou energii a je tedy nutné se ji nějakým způsobem zbavit, jinak může dojít k poškození tranzistorů tímto zbytkovým napětím. Toto se řeší přidáním diod v opačném směru od tranzistorů, kterými vznikne cesta, kterou lze elektrické energie spotřebovat.

## 4.4 Moduly

V této podkapitole jsou popsány moduly, pomocí kterých je tvořena modifikovaná hlavice a na jakých principech fungují.

### 4.4.1 NodeMCU

NodeMCU[8] je open-source hardwarová, tak softwarová platforma. Dokumentace k deskám je volně dostupná a je tedy možné pro každého vyrobit si vlastní desku. Nejčastěji je možné se setkat s NodeMCU od výrobců Amica a LoLin, avšak obě firmy vyrábějí v velice podobný výrobek. NodeMCU nabízí jedenáct digitálních vstupně-výstupních pinů, z nichž deset je schopno použít pulzně šířkovou modulaci, a jeden analogový pin, který lze použít jako A/D převodník, jehož rozpětí je 0 - 1V. Mezi další periférie patří UART, SPI a I2C rozhraní. Je možné jej napájet přes micro USB, přes který se také nahrává program pro vykonávání, nebo přes pin  $V_{in}$ , který lze vidět na obrázku 4.5. Hlavními částmi NodeMCU je mikrokontrolér ESP8266 s podporou WiFi 802.11 na 2.4 GHz. 4MB flash paměť a RAM o obsahu 80KB. Mikrokontrolér pracuje s operační rychlostí 80 MHz, obsahuje 4MB flash paměti, z které je možno použít až 1MB na uložení programu, 64KB SRAM paměti.

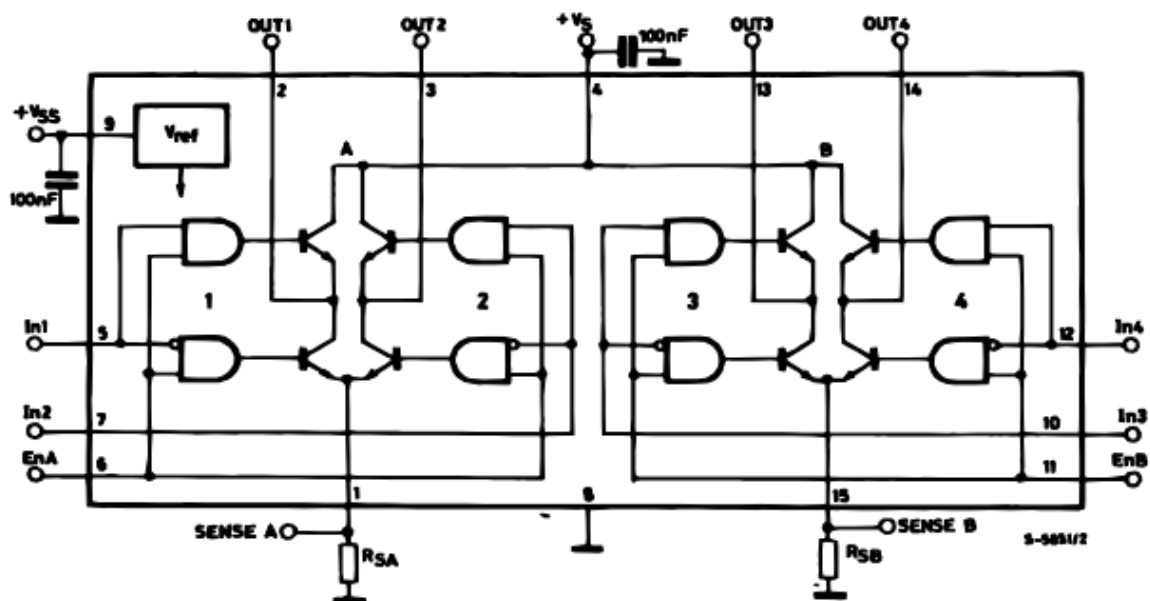


Obrázek 4.5: Schéma pinů na NodeMCU V3<sup>5</sup>

#### 4.4.2 Modul L298N

L298N (viz schéma modulu B.1) je modul s čipem L298 s dvěma H-můstky (viz schéma čipu na obrázku 4.6), se kterým tedy lze ovládat dva stejnosměrné motory. Logiku modulu je možné napájet pěti až dvanácti volty, které jsou však regulátorem sníženy na pět voltů pro logiku, a pokud motory potřebují více než deset voltů (až třicet pět voltů), tak je lze napájet zvlášť dalším pinem. Modul má napěťový úbytek dva volty, takže je vždy nutné napájet potřebným napětím plus dva volty navíc. Oba motory mají tři vstupní piny, dva pro ovládání směru a jeden pro ovládání rychlosti, který může být propojený k sousednímu povolovacímu pinu (znázorněno krátkým žlutým propojením na modulu L298N na obrázku C.1), na kterém se nachází napětí od zdroje, které signalizuje maximální rychlost.

<sup>5</sup>Převzato z <https://www.neven.cz/kategorie/elektronicke-soucastky/elektronicky-vyvoj/vyvojove-desky/ostatni/nodemcu-lua-amica-r2-esp8266-wifi-vyvojova-deska/>



Obrázek 4.6: Schéma obvodu čipu L298<sup>6</sup>

#### 4.4.3 Senzor DHT22

Digital-output relative humidity and temperature sensor (DHT) je senzor umožňující měření relativní vlhkosti pomocí polymerového kondenzátoru a aktuální teploty pomocí NTC termistoru. Senzor DHT22 je někdy označován jako AM2302, jedná se o totožný senzor, AM2302 má místo pinů pro vstupy a výstupy drátky. Senzor obsahuje i jednoduchý čip pro spolehlivý převod hodnot z analogových na digitální. Modul poskytuje dva piny pro napájení, jeden pro zdroj a jeden pro zem, a jeden datový pin pro získávání hodnot. Vstupní napětí senzoru by mělo být tři až pět voltů. Datový výstup využívá digitální komunikace, aby bylo možné číst jak teploty, tak vlhkost. Existuje velká řada volně dostupných knihoven, které umožňují jednoduchou práci s tímto senzorem. Vzorkovací frekvence senzoru je 0.5Hz a tudíž lze hodnoty číst pouze jednou za dvě sekundy. Rozmezí měřené teploty je -40 až 80°C s chybou  $\pm 0.5$  °C, vlhkost je možné měřit v plném rozmezí s chybou 2 až 5%.

### 4.5 Modifikace

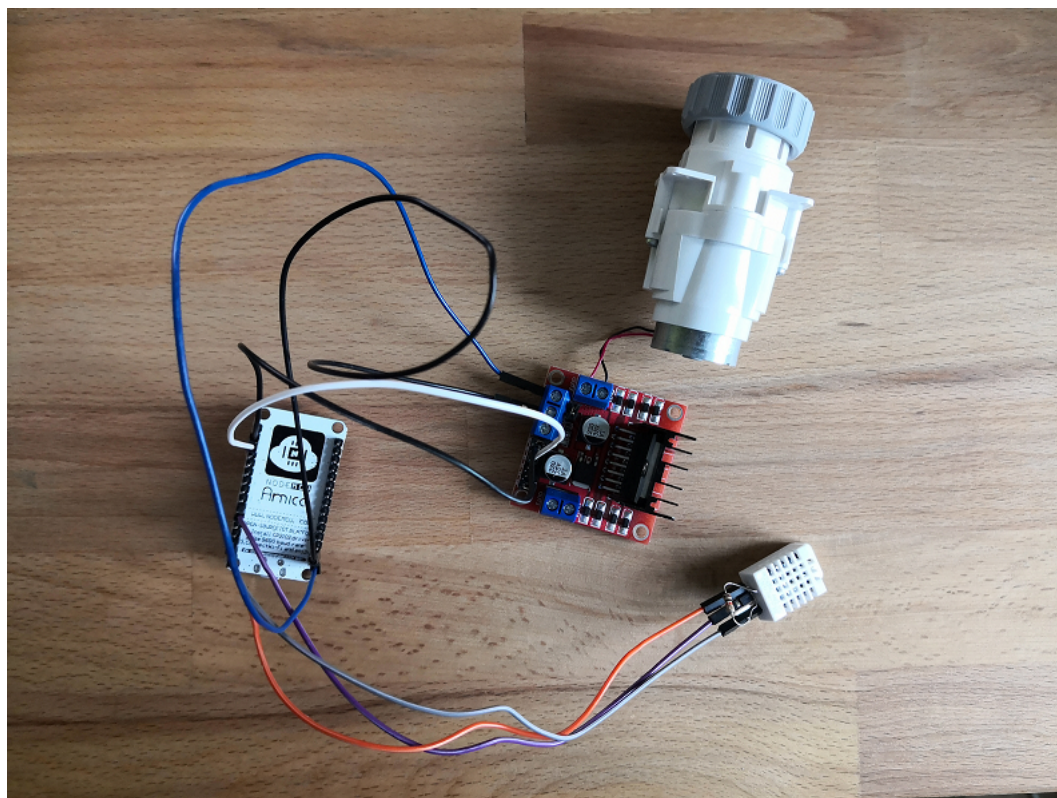
Prvním úkonem modifikace bylo otevření hlavice eQ-3 N. Dále bylo potřeba odstranit řídicí elektroniku, odpojit stejnosměrný motorek od řídicí elektroniky a odstranit zásobník pro baterie. Po těchto úkonech zůstal jen motorek s aparátem, kterým pohybuje, který slouží k ovládání ventilu topení.

Pro správný provoz senzoru DHT22, je třeba přivést na něj napětí 3 až 6 voltů, v tomto případě je senzor napájen z pinu NodeMCU 3V3, který přivádí napětí 3.3 voltů. Je nutné

<sup>6</sup>Převzato z [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)



Pro chod motoru je nutné přivést do modulu L298N napětí o velikosti pěti voltů, což je o dva více, než potřebuje pro svůj chod motor, z důvodu ztráty napětí v modulu L298N. Pět voltů lze na NodeMCU získat z pinu  $V_{in}$ . Modul musí být připojen k zemi. Pro ovládání motoru je potřeba zapojit digitální piny z NodeMCU na vstupní piny modulu, a připojit stejnosměrný motor k výstupu modulu. Rychlost motoru by se dala ovládat, kdyby se odstranila propojka na povolovacím pinu, která na něm v provozu drží vysoké napětí, což značí nejvyšší rychlost, a místo propojky by se připojil digitální pin s podporou pulzně šířkové modulace, nebo analogový pin. Příklad výsledného zapojení je vyobrazen na obrázku 4.7.



27

## Kapitola 5

# Návrh systému pro řízení termostatických hlavice

Tato kapitola popisuje návrh částí systému pro řízení termostatických hlavice. Jako první je popsán návrh serverové části, která slouží jako úložiště dat a jejím úkolem je tyto data poskytovat ostatním prvkům systému. Dále je rozebrán návrh lokálního řízení samotné termostatické hlavice a jakým způsobem je realizováno řízení vytápění. A poslední část se věnuje popisu návrhu grafického rozhraní, pro zadávání požadavků na řízení uživatelem.

### 5.1 Serverová část systému

Serverová část systému slouží ke komunikaci mezi jednotlivými prvky systému a k uchování dat, které prvky potřebují znát, nebo si je mezi sebou vyměňují.

Server pracuje na základě protokolu HTTP a je navržen pomocí architektury REST [5]. Dle původního návrhu je jádrem serverového modulu třída `RequestHandler`, která obsahuje jednoduchý server, přijímající HTTP požadavky. Pro filtraci požadavků třída implementuje jednotlivé metody pro obsluhu čtyř základních HTTP požadavků (GET, POST, PUT, DELETE). Pro další filtraci je použit návrhový vzor tovární metoda, do které server přepośle přijatou zprávu. Tovární metoda má za úkol vytvořit objekt zprávy na základě cíleného endpointu zprávy (viz obrázek 5.1). Tovární metoda pošle vytvořený objekt zprávy třídě vydavatele `ValveKeeper` (návrhový vzor pozorovatel), který oznámí všem objektům třídy `ThermostaticValve`, reprezentující jednu hlavici, které jsou k němu přihlášeny k odběru, zprávu o změně stavu společně s objektem zprávy. Objekt hlavice vykoná akce příslušné k dané zprávě, pokud ji byla určena.

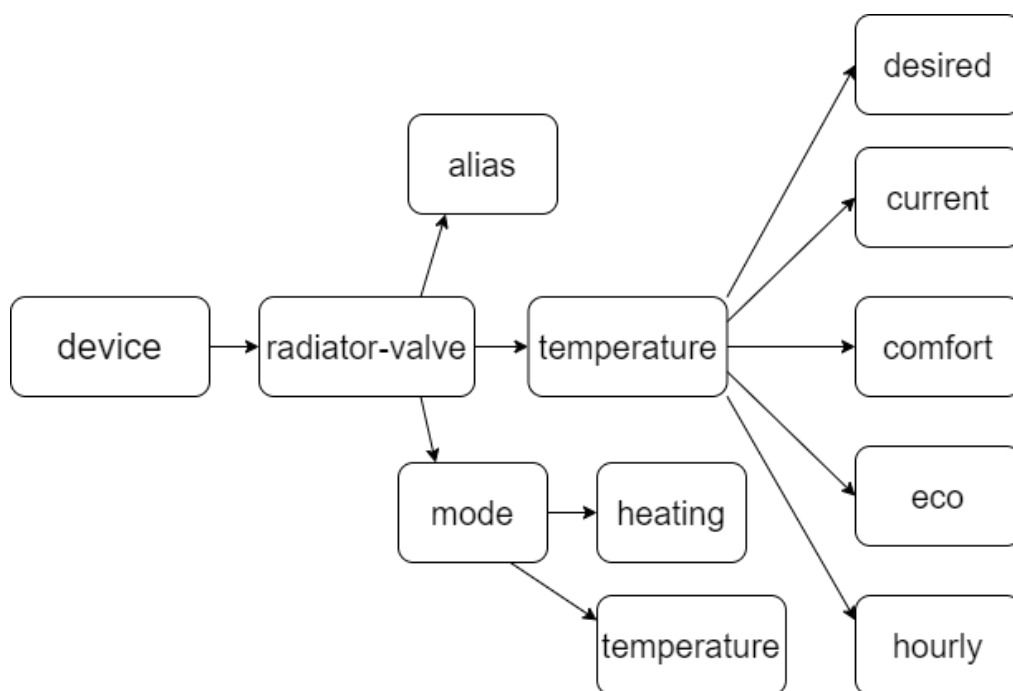
Podle aktualizovaného návrhu je jádrem modulu HTTP server, postavený na knihovně Flask [10]. Knihovna umožňuje nadefinovat rozdílné metody pro specifikované cílové cesty (endpoints), tudíž lze eliminovat třídu s návrhovým vzorem tovární metody, jejímž úkolem byla filtrace požadavků. Po přijetí požadavku knihovnou Flask, se požadavek přepośle třídě `Server`, který jej označí vhodným identifikátorem a přepośle jej třídě `ValveKeeper`, která vykoná stejné akce jako v původním návrhu.

Objekt hlavice si uchovává informace o nastavených teplotách pro rozdílné módy. Mezi módy patří eko, komfortní a vytápění dle časového plánu. Dále si uchovává skutečnou teplotu dané hlavice pro případné statistiky.

Endpoint `\device` je v aktuálním kontextu redundantní, ale zajišťuje rozšiřitelnost systému



o další zařízení jiného typu než je termostatická hlavice. Přes `\device\radiator-valve` lze vytvořit objekt hlavice a získat obecné základní informace ohledně stavu jedné hlavice. Alias hlavice (pojmenování hlavice uživatelem pro přehlednost) lze získat a nastavovat přes `\device\radiator-valve\alias`. K vytápěcímu módu hlavice je možné přistupovat přes cestu `\device\radiator-valve\mode`, tento endpoint se dále dělí na `\heating`, reprezentující jaký algoritmus má být použit pro řízení vytápění, a `\temperature`, reprezentující jaký teplotní mód je aktivní a měla by se tedy použít jeho hodnota. K nastaveným a aktuálním teplotám lze přistupovat pomocí `\device\radiator-valve\temperature`, dále lze přistupovat k jednotlivým nastavitelným teplotám jako je `eco`, `hourly` a `comfort`, `current` je hodnota aktuální teploty a `desired` je dynamická hodnota, která záleží na aktuálním vytápěcím módu.



Obrázek 5.1: Endpointy reprezentovány stromovou strukturou

## REST

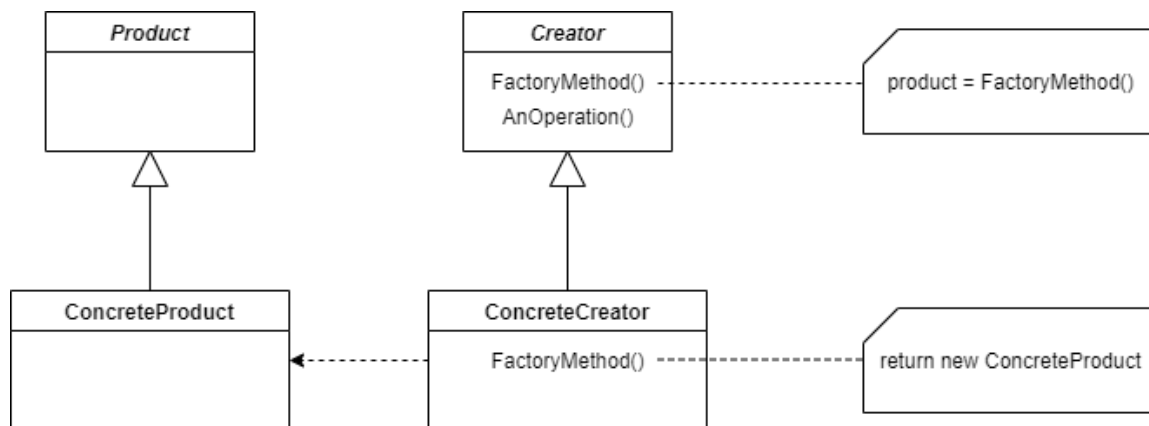
REST (REpresentational State Transfer) je datově orientovaná architektura veřejného rozhraní, nedefinuje tedy procedury, ale cesty k datům, pomocí endpointů. REST nejčastěji pracuje pomocí protokolu HTTP (HyperText Transfer Protocol), ale lze jej implementovat i pomocí jiných protokolů. REST využívá čtyři základní metody modelu CRUD (Create, Retrieve, Update, Delete), v HTTP jsou tyto metody POST, GET, PUT a DELETE. Hlavním účelem je vytvořit rozhraní, ke kterému bude možné realizovat přístup z rozdílných platforem, díky použití rozšířeného protokolu.

### Návrhový vzor tovární metoda

Úkolem tovární metody [11] je vytváření rozdílných produktů se stejnými základními atributy, které dědí z rozhraní abstraktního produktu, pomocí jednoho rozhraní/třídy (viz diagram na obrázku 5.2).

Tovární metoda je implementována ve třídě, kde metoda na základě parametrů vytvoří daný objekt, který však sdílí rozhraní se všemi vytvořitelnými objekty touto metodou. Tovární metoda může být také implementována jako rozhraní, ze kterého dědí další třídy, kde každá třída vytváří děděnou tovární metodou rozdílný produkt.

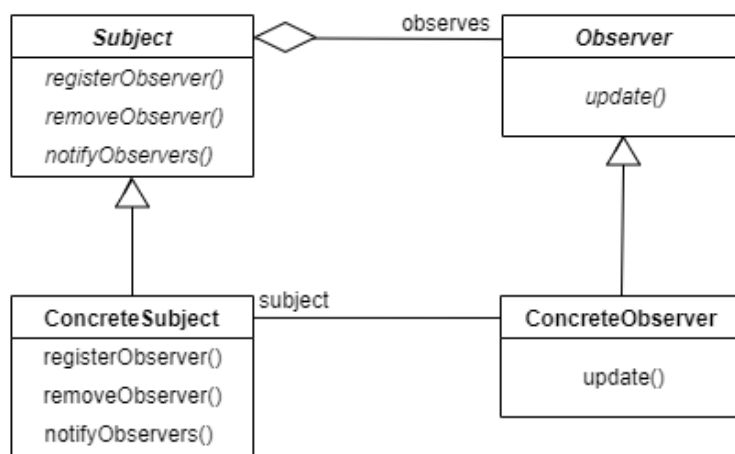
S každým objektem lze tedy pracovat jednotně bez znalosti o jaký specifický objekt se jedná.



Obrázek 5.2: Diagram návrhového vzoru tovární metody

### Návrhový vzor pozorovatel

Pozorovatel [11] se skládá ze dvou objektů, vydavatele a předplatitele (pozorovatele). Předplatitel má možnost přihlásit se k odběru u vydavatele. Vydavatel obsahuje stav, při nějaké změně tohoto stavu vydavatel oznámí všem svým přihlášeným předplatitelům, že došlo ke změně, předplatitelé změnu zaregistrují a použijí ji ve vlastní logice (viz diagram na obrázku 5.3).



Obrázek 5.3: Diagram návrhového vzoru pozorovatele

## 5.2 Řízení termostatické hlavice

Hlavice si při spuštění načte MAC adresu svého čipu a druhou polovinu adresy použije jako svůj unikátní identifikátor v rámci systému řízení hlavic, toto je možné protože jsou hlavice navrženy tak, že budou používat čipy ESP8266 od výrobce Espressif Systems, tudíž by všechny čipy měly mít druhou polovinu MAC adresy rozdílnou (první polovina MAC adresy udává výrobce a druhá by měla být v rámci výrobce unikátní). Dále se připojí na zadanou síť WiFi a pošle na server požadavek POST na vytvoření hlavice s jejím identifikátorem.

Za běhu se hlavice každých dvanáct minut dotáže serveru, zda nedošlo ke změně řízení, zda se nezměnila požadovaná teplota a algoritmus pro řízení teploty, a odešle na server aktuální naměřenou teplotu, a každé tři minuty získává naměřenou teplotu senzorem a porovnává ji s požadovanou teplotou a kontroluje zda nemá dojít ke změně pozice ventilu.

## 5.3 Grafického uživatelské rozhraní

Grafické rozhraní (viz mockup na obrázku 5.4) slouží uživateli pro zadávání požadavků na způsob vytápění a sledování průběhu vytápění. Uživatel pomocí něj může ovládat a sledovat všechny hlavice v systému. Sledovat u hlavic může uživatel aktuální teplotu hlavice a nastavenou požadovanou teplotu. Lze nastavovat požadované teploty u všech módů a lze nastavit alias, který je zobrazen ve výběru hlavic. Která teplota se bude brát jako požadovaná, udává nastavený mód, a jaký vytápěcí mód se použije (PID nebo hysterezní) udává nastavený vytápěcí mód.

Grafické rozhraní komunikuje se serverovou částí pomocí HTTP požadavků, které stojí na architektuře REST, tímto se dosáhne, že grafické rozhraní nebude závislé na stroji, kde bude server běžet, ale bude možné jej ovládat i z rozdílných zařízení.

Thermostatic Valves GUI	
<b>Valves list</b>	
id: 649 (living room)	
id: 35 (kitchen)	
id: 197 ()	
id: 992 (bedroom)	
Current temperature: 21.3°C	
Desired temperature: 22°C	
Comfort temperature:	22
Eco temperature:	18
Day: Mon	Hour: 17
Hourly temperature: 21	
Alias:	living room
Submit	
Mode:	Comfort
Heating mode:	hysteresis
Time: 17:53	

Obrázek 5.4: Mockup grafického rozhraní

## Kapitola 6

# Implementace systému pro řízení termostatických hlavice

V této kapitole je popsán způsob implementace jednotlivých navržených modulů pro podporu řízení modifikovaných hlavice. V první části kapitoly je zmíněna implementace serveru pro obsluhu požadavků z termostatické hlavice a od uživatele. Následuje popis implementace termostatické hlavice. A dále je rozebrán způsob implementace grafického rozhraní.

### 6.1 Serverová část

Serverová část je implementována v programovacím jazyce Python. Implementace se řídí podle návrhu v předešlé kapitole 5. HTTP server je implementován za použití knihovny Flask, což je webový framework pro jazyk Python.

Flask aplikace se vytvoří tak, že se vytvoří objekt Flask, na kterém se poté dají definovat různé HTTP cesty (routes) daného API i se specifikováním typu požadavku, který se spouští na zadané IP adrese a portu (viz ukázka 6.1).

```
import flask

app = flask.Flask(__name__)

@app.route("device/radiator-valve/temperature/desired", methods=["GET"])
def get_desired_temperature():
    """code logic"""
    return response_body, response_code

app.run(host="1.2.3.4", port="12345")
```

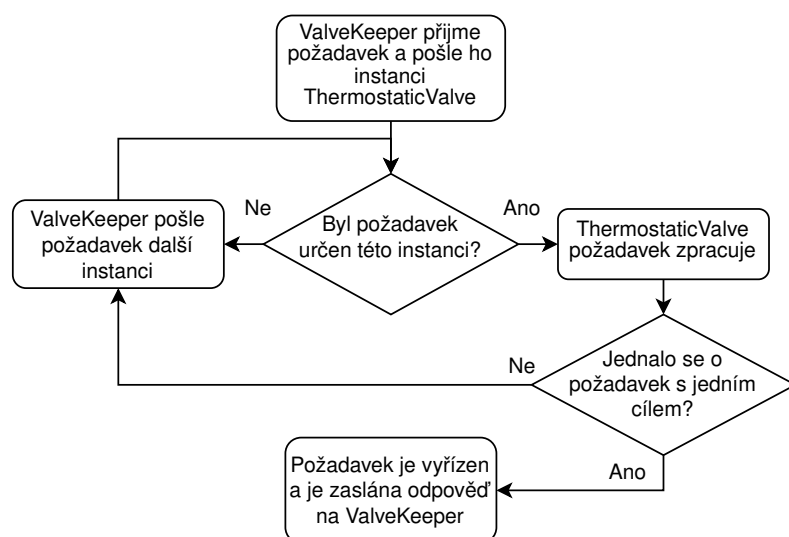
Výpis 6.1: Příklad inicializace Flask objektu a cesty

Modul ve kterém běží Flask po přijetí zprávy deleguje tuto zprávu instanci třídy **Server**, z důvodu že Flask aplikace dle dokumentace by se měla nacházet pouze v modulu a neměla by být zaobalena v jiné třídě. Ačkoliv existují knihovny, které umožňují vytvoření aplikace a jejích cest uvnitř třídy, rozhodl jsem se pro delegaci do jiné třídy.

Flask modul volá pro každý druh požadavku rozdílnou funkci třídy **Server**, které předá **Request** objekt, který obsahuje globální informace o aktuálně obsluhovaném požadavku,

jako jsou argumenty požadavku nebo naformátované tělo zprávy. Atributem třídy `Server` je `keeper`, který obsahuje instanci třídy `ValveKeeper`, na kterou deleguje požadavek předáním objektu `Request` a řetězce identifikující typ požadavku, pokud se jedná o požadavek na získání nebo změnění informací o nějaké hlavici. Pokud se jedná o požadavek na vytvoření, odstranění hlavice, nebo na získání identifikátorů dostupných hlavíc, tak tuto operaci zajišťuje samotný `Server`.

Instance třídy `ValveKeeper` obsahuje množinu instancí třídy `ThermostaticValve`, reprezentující jednotlivé hlavice v systému, tyto hlavice jsou v roli předplatitelů vůči instanci `ValveKeeper`, který je v roli vydavatele. Když `Server` přepośle požadavek na `ValveKeeper`, tak ten jej přímo přepośle všem instancím `ThermostaticValve` z množiny, nebo pouze těm, které předcházeli v nalezení té, které byla zpráva určena. `ThermostaticValve` se po přijetí požadavku podívá, zda je specifikován parametr `id`, který identifikuje jednotlivé instance `ThermostaticValve`, pokud je specifikován, tak se porovná s identifikátorem dané instance. Pokud se identifikátory nerovnájí, tak `ValveKeeper` posílá požadavek dalšímu předplatiteli (viz vývojový diagram 6.1).



Obrázek 6.1: Vývojový diagram doručování požadavku cílové instanci `ThermostaticValve`

Pokud se jedná o požadavek určený jedné cílové instanci (obsahuje parametr `id`), tak po jeho doručení cílové hlavici, hlavice vrací hledanou hodnotu (pouze v případě získávání dat), stavový kód HTTP a `return_code` specifikující, zda byl požadavek doručen. Pokud byl požadavek doručen cíli, tak `ValveKeeper` získanou hodnotu a HTTP kód vrací serveru, který tyto hodnoty vrátí modulu s Flaskem, který tělo zprávy přeloží do formátu JSON a odešle je klientovi, který tento požadavek podal. Pokud se však jedná o požadavek bez specifikovaného cíle, tak je tento požadavek postupně oznamován všem hlavicím, při prvním úspěšném oznámení (zprávu dostala jedna z hlavíc, které byla určena) se `ValveKeeper` dozví, že bude moci odeslat pozitivní HTTP kód klientovi. Po oznámení požadavku všem hlavicím je postup stejný, jako u požadavku se specifikovaným cílem.

Pokud se jedná o požadavek pracující pouze s jednou hodnotou, tak se v JSON formátu jedná o řetězec nebo číslo. Pracuje-li se s jedním typem hodnoty, ale u více hlavíc, tak se pracuje s listem, což je v JSON formátu pole, v případě, kdy není potřeba znát vlastníka dané hodnoty (například při získávání identifikátorů hlavíc v systému). Pokud je vlastníka

potřebné znát, tak se pracuje s python slovníkem, který je po překladu ve formátu JSON objektu, kde klíčem k atributu je identifikátor hlavice.

Při práci s několika rozdílnými atributy je využít slovník s označením atributu jako klíčem k hodnotě, tedy s JSON objektem. Když je potřeba pracovat s rozdílnými atributy u více hlavic, tak je použit zanořený slovník, kde v roli klíče je identifikátor, který vede k dalšímu slovníku, ve kterém se nacházejí označení atributů s jejich hodnotami, v JSON formátu se jedná o zanořený JSON objekt (ukázka výpisu 6.2).

```
{
  "176":{
    "comfort":21.0,
    "eco":18.5,
    "mode":1
  },
  "359":{
    "comfort":22.0,
    "eco":19.0,
    "mode":2
  }
}
```

Výpis 6.2: Příklad zprávy v JSON formátu s více hodnotami od více hlavic

## 6.2 Řízení termostatické hlavice

Řízení termostatické hlavice je implementováno přes modul NodeMCU, pomocí vývojového prostředí Arduino<sup>1</sup> a programovacího jazyka Arduino, založeném na jazyce C++.

Při spuštění mikrokontroléru se první definují a deklarují globální proměnné, jako jsou například potřebné informace k připojení k WiFi síti, či cílová adresa serveru. Pin z něhož se budou získávat data ze senzoru DHT se inicializuje za pomoci Adafruit knihovny *DHT – sensor – library*<sup>2</sup>, vytvořením objektu DHT, kterému je třeba při definici předat parametry o typu senzoru DHT, neboť knihovna podporuje ovládání všech senzorů typu DHT, DHT 11, DHT 21 (AM2301), DHT 22 (AM2302) a AM2321, při návrhu byl použit senzor DHT22, je tedy třeba zadat jako parametr DHTTYPE hodnotu DHT22, dále je nutné specifikovat pin, ze kterého se budou za chodu programu získávat naměřené hodnoty, zvoleným pinem je D7, který se nachází ve stabilních stavech při inicializaci nebo resetu modulu NodeMCU.

Následující implementace se skládá z dvou hlavních funkcí – **setup** a **loop**. Funkce **setup** probíhá pouze jednou během chodu programu, po spuštění modulu a nebo po jeho resetu. Tato funkce má za úkol inicializaci stavu modulu a jeho prvků. Po jejím dokončení se začne cyklicky provádět funkce **loop** po zbytek běhu mikrokontroléru, která obsahuje hlavní logiku prováděného programu.

Ve funkci **setup** je třeba zprvu inicializovat jednotlivé piny, ovládající připojené moduly jako čtecí nebo zapisovací. Pin D1 a D2 se nastaví jako zapisovací, protože se jimi bude

<sup>1</sup><https://www.arduino.cc/en/software>

<sup>2</sup><https://github.com/adafruit/DHT-sensor-library>

ovládat směr pohybu motoru. Pin D7 se nastaví na čtecí, z tohoto pinu se budou číst hodnoty z teplotního senzoru. Dále je za pomoci knihovny *ESP8266WiFi* a funkce `macAddress` získána MAC adresa čipu ESP8266, jejíž druhá polovina bude použita pro unikátní identifikaci dané hlavice v systému. Následně začne připojování v zadané síti WiFi. Po připojení se mikrokontrolér pokusí odeslat POST požadavek na vytvoření dat o dané hlavici na server, pomocí knihovny *ESP8266HTTPClient*. Příchozí stavový kód HTTP s číslem 200 značí, že se hlavice s tímto identifikátorem již v systému nachází, stavový kód 201, značí úspěšné vytvoření hlavice v systému, v obou těchto případech program zapíše do booleovské proměnné `posted` hodnotu `true` a pokračuje dále, v případě jiného stavového kódu se pokusí požadavek poslat ještě devětkrát, ale poté i při neúspěchu pokračuje dále. Jako poslední ve funkci `setup` se načte 128 bytů paměti EEPROM (Electrically Erasable Programmable Read-Only Memory), použitím funkcí knihovny *EEPROM*<sup>3</sup>, ačkoliv modul NodeMCU EEPROM paměť neobsahuje, knihovna je schopná ji emulovat na paměti flash, knihovna je schopna emulovat až 4096 bytů paměti EEPROM. Po načtení se přečtou 99., 100. a 101. byte (byty se zde číslují od nuly), obsahuje-li 99. a 101. byte hodnotu 111, tak se na 100. bytu nachází aktuální pozice ventilu, která se načte do proměnné, která indikuje pozici ventilu, pokud 99. a 101. byte neobsahuje hodnotu 111, tak se jedná o prvotní aktivaci hlavice a je třeba ventil dostat do nulové pozice, tudíž se aktivuje motorek, aby se ventil úplně uzavřel. Po této akci končí funkce `setup`.

Ve funkci `loop` se při prvním cyklu ihned získá naměřená hodnota z teplotního senzoru, která se poté uloží do pole posledních tří naměřených teplot. Poté zkontroluje zda nemá nastat změna pozice ventilu funkcí `control_valve`, funkce si první zkontroluje, zda za poslední tři měření neklesla teplota o 1.0 stupňů Celsia (pouze pokud jsou k dispozici tři poslední měření, tedy nejdříve až po šesti minutách chodu), což by indikovalo otevření okna a ventil by se na 6 minut zavřel, po této době by se ovládání vrátilo do normálu. Pokud otevřené okno nebylo detekováno, tak se nastaví ventil podle aktuálně zvoleného vytápěcího algoritmu. Je-li nastaven hysterezní algoritmus, tak se ventil otevře, pokud je teplota nižší než žádaná mínus hraniční hodnota hystereze a teplota je ve stavu klesání, pokud je teplota ve stavu vzestupném a překročila žádanou teplotu plus hraniční hodnotu, tak se ventil zavře. Jestliže je nastaven algoritmus PID regulátoru, je aplikována rovnice

$$Kp * error + \sum_{n=0}^x (Ki * error_n * time) + Kd * (temp_i - temp_{i-1}) / time \quad (6.1)$$

kde hodnoty  $K$  značí jednotlivé složky regulátoru, *error* je rozdíl mezi žádanou a aktuální hodnotou,  $temp_i$  je aktuální teplota a *time* je doba v minutách mezi měřeními, výsledek je zaokrouhlen na celé číslo a ořezán do intervalu  $\langle 0; 30 \rangle$ , ten se poté použije pro nastavení pozice hlavice. Po získání hodnoty pro nastavení ventilu se nastaví signály pro řízení motoru na potřebnou dobu, tak aby bylo dosaženo získané pozice. Když se nejedná o první cyklus, tak se tato sekvence úkonů opakuje každou jednu minutu.

Dále pokud je proměnná `posted` nastavena jako pravdivá, tedy hlavice byla úspěšně zaregistrována do systému, tak se odesílá na server aktuálně změřená teplota. Hodnota se posílá jako požadavek typu PUT ve formátu JSON, který je získán za použití knihovny *ArduinoJson*<sup>4</sup>. Následně se pokusí získat hodnotu žádané teploty ze serveru požadavkem GET a poté zkusí získat nastavený vytápěcí algoritmus a důležité hodnoty k jeho použití (hysterezní hraniční hodnotu a koeficienty všech složek pro algoritmus PID) požadavkem

<sup>3</sup><https://github.com/esp8266/Arduino/tree/master/libraries/EEPROM>

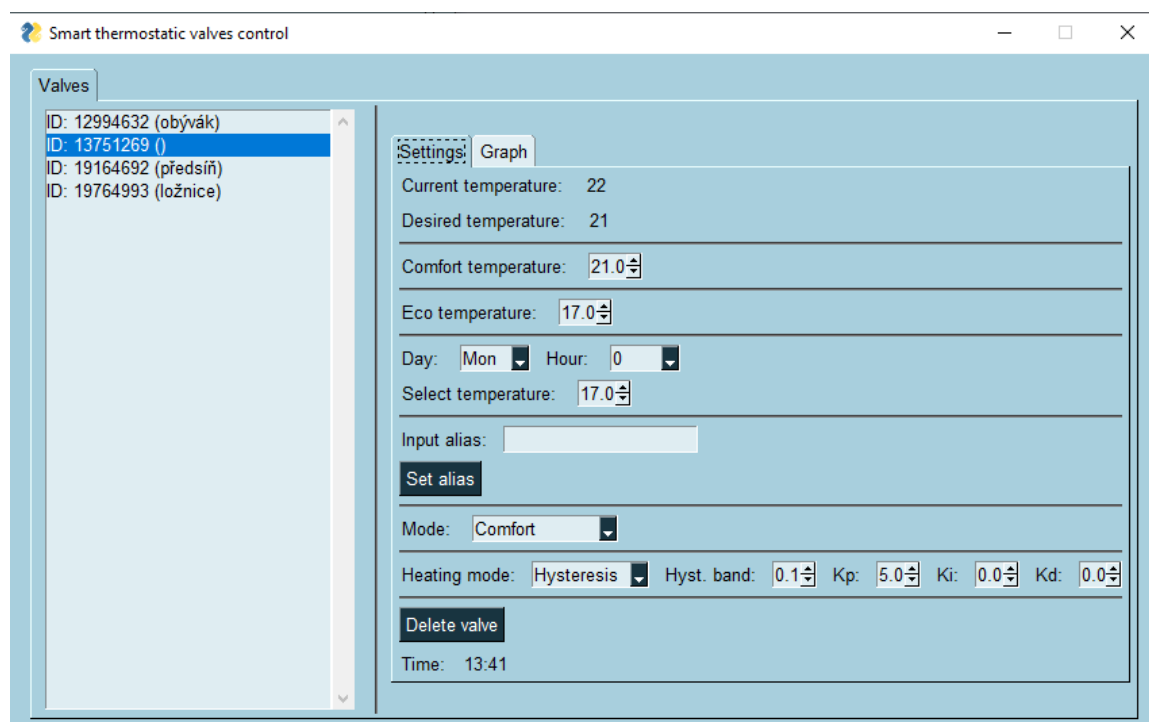
<sup>4</sup><https://github.com/bblanchon/ArduinoJson>

GET (jestliže nastal neúspěchu požadavků, tak se hodnota těchto hodnot nemění). V případě, že hodnota proměnné `posted` je nepravdivá, tak se pokusí jednou odeslat požadavek typu POST za účelem vzniku dat o hlavici na serveru. Tato sekvence příkazů se opakuje každých deset minut.

## 6.3 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní je implementováno v programovacím jazyce Python, za použití knihovny *PySimpleGUI*<sup>5</sup>.

Hlavní stránka rozhraní je rozdělena na dvě části (viz obrázek 6.2). V první části se nachází seznam se zaregistrovanými hlavicemi v seznamu, kde se hlavice dají odlišit dle unikátního identifikátoru v systému, nebo podle aliasu zadaného uživatelem, který je defaultně prázdný řetězec. Druhá sekce se skládá ze dvou záložek, kdy obě nabízejí uživateli přehled informací o hlavici. První záložka obsahuje přehled základních nastavení a elementy umožňující jejich úpravu, konkrétně se v první části záložky nachází poslední naměřená hodnota, která byla odeslána serveru, a pod ní požadovaná teplota. Následují tři prvky pro změnu požadovaných teplot u rozdílných programů, kde změna teploty je realizována za použití tak zvaného spinneru. Dále je nastavení aliasu pomocí textové vstupu a potvrzujícího tlačítka. Změna módu pro využití požadované teploty, tak vytápěcího algoritmu je realizována listboxem, který obsahuje všechny možné hodnoty. Informace k vytápěcímu algoritmu je možné měnit pomocí již zmíněných spinnerů.



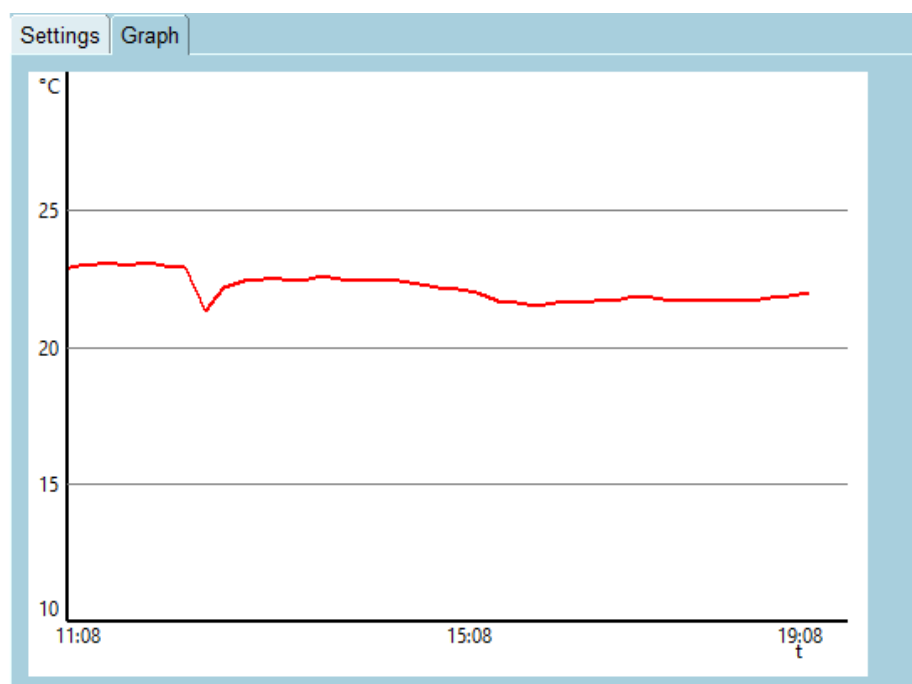
Obrázek 6.2: Výsledný vzhled grafického uživatelského rozhraní

Druhá část informační sekce obsahuje spojnicový graf (viz obrázek 6.3), který vyobrazuje pohyb teploty, podle naměřených hodnot v posledních osmi hodinách. Svislá osa

<sup>5</sup><https://github.com/PySimpleGUI/PySimpleGUI>



určuje zobrazovanou teplotu s hranicemi na deseti a třiceti stupních Celsia. Vodorovná osa určuje čas zobrazené hodnoty. Hlavičky odesílají naměřené hodnoty každých dvanáct minut, tudíž maximální počtem zobrazených hodnot je čtyřicet.



Obrázek 6.3: Část grafického uživatelského rozhraní s grafem teploty

Pro komunikaci se serverem je využita knihovna *requests*<sup>6</sup>. Pro získávání hodnot je využit požadavek typu GET, pro změnu hodnot na serveru se používá typ PUT a pro mazání dat DELETE.

<sup>6</sup><https://docs.python-requests.org/en/master/>

## Kapitola 7

# Testování

Obsahem této kapitoly je způsob, jakým byly testovány jednotlivé moduly a byla ověřena správnost jejich chování. Správné chování serveru bylo ověřeno jednotkovými testy na třídu `ThermostaticValve` a integračními testy bylo ověřeno chování jako celku. Chování samotné termostatické hlavice bylo ověřeno dlouhodobými, ale i kratšími testy běhu.

### 7.1 Testování serveru

Serverová část byla testována pomocí jednotkových testů, kterými byly ověřeny výstupy metod a chování třídy `ThermostaticValve`, která obsahuje zásadní část logiky této části. Integrační testy probíhají nad běžícím serverem, na který posílají různé požadavky a ověřují správné zpracování dat a odpovědi na požadavky. Testy této části byly implementována za použití Python knihovny *unittest*.

### 7.2 Testování termostatické hlavice

Termostatická hlavice byla testována několika zkušebními běhy, kde test byl zaměřen na specifickou funkci. Proběhlo několik manuálních testů na ověření správného získávání a zasílání dat na server. Byly provedeny testy na detekci otevřeného okna, u kterých bylo zjištěno, že nezáleží pouze na venkovní teplotě, ale také na vzdálenosti hlavice od okna, směru větru a venkovním tlaku (při špatném větru či nízkém tlaku nedochází k proudění vzduchu z vnějšku dostatečně rychle). Při příznivějších podmínkách funguje detekce otevřeného okna do přibližně 17°C.

Pro oba algoritmy bylo provedeno několik testových běhů, které se lišily ve vnějších podmínkách, nastavených programech a teplotách. Delší běhy sdílely nastavený program, který byl časový, v průběhu dne od 9:00 do 19:59 byla nastavena teplota 22°C a od 20:00 do 8:59 teplota 19°C (viz 7.1).

Délka	Algoritmus	Vnější teploty	Rozdíl od požadované
3 dny	PID (Kp:50, Ki:0, Kd:0)	noc: -3°C, den: 8°C	-0.2°C až -0.5°C
2 dny	PID (Kp:50, Ki:0.1, Kd:0)	noc: 0°C, den: 9°C	-0.2°C až 0.1°C
3 dny	Hysterezní (pás=0.1)	noc: 2°C, den: 8°C	-0.3°C až 0.2°C
2 dny	Hysterezní (pás=0.3)	noc: 0°C, den: 7°C	±0.5°C

Tabulka 7.1: Detaily jednotlivých delších běhů

Při testování hysterezního algoritmu, tak bylo zjištěno, že při nastavení hysterezní hranice na  $0.1^{\circ}\text{C}$ , tak vytápění bylo spolehlivé a oscilovalo okolo požadované teploty s maximální chybou přibližně okolo  $0.3^{\circ}\text{C}$  (viz příklad grafu z jedno z běhů 7.1).



Obrázek 7.1: Graf s průběhem teploty z testování hysterezního algoritmu

Při použití algoritmu PID regulátoru pouze s proporcionální složkou nenulovou na hodnotě 50, bylo řízení přesnější, ale teplota kolísala přibližně  $0.2$  až  $0.5^{\circ}\text{C}$  pod požadovanou hodnotou. Po přenastavení integrační složky na hodnotu 0.1, teplota oscilovala velmi přesně okolo požadované hodnoty, po překonání začáteční nestability a přechodu ke stabilnímu stavu.

## Kapitola 8

# Závěr

Primárním cílem této bakalářské práce bylo navrhnout a provést modifikaci levné digitální termostatické hlavice na chytrou řízenou modulem založeným na ESP8266, využívající komunikaci přes síť WiFi. Dále navrhnout a implementovat systém pro řízení těchto termostatických hlavice. A nakonec systém otestovat zda splňuje cíle zadané prací.

V první části práce je detailně popsána funkcionalita, kterou hlavice mohou nabízet, dále je přiblížen rozdíl mezi digitálními a chytrými hlavicemi a nakonec jsou popsána některá dostupná řešení, ať už komerční, či zdarma dostupné návrhy. Další část je zaměřena na popis algoritmů, které jsou využívány pro regulaci teploty v lokálních ovládacích prvcích vytápěcího systému (jako například termostatické hlavice), je zde zmíněn algoritmus využívající hysterezní smyčky a pokročilejší algoritmus PID regulátor. V následující části je rozebrána modifikace vybrané termostatické hlavice a technologie společně s moduly, které jsou potřeba pro tuto modifikaci. V poslední část se zabývá návrhem a implementací celého systému pro řízení hlavice, a následně jeho testováním.

Byl vytvořen prototyp modifikované termostatické hlavice, založené na komerčně dostupné hlavici eQ-3 N. Hlavice dokáže fungovat samostatně, ale pro její ovládání je potřeba spuštěný vytvořený server, založený na architektuře REST, ze kterého by získávala požadovaná data uživatelem. Uživatel může tento systém ovládat za použití vytvořeného grafického rozhraní, které komunikuje se serverem, kterému posílá požadované data a také z něj získává aktuální data. Lze pomocí něj sledovat aktuální nastavení jednotlivých hlavice systému, a také jejich aktuální stav.

Díky architektuře REST, na které je server postaven, je zajištěna jednoduchá rozšiřitelnost systému o nová zařízení chytré domácnosti.

Systém je navrhnout a implementován tak, aby umožnil jednoduché přidání dalších zařízení. V rámci budoucích rozšíření by byl vhodný přechod na komunikaci přes rádiové vlny frekvence 868 MHz, což by snížilo spotřebu a s dalšími úpravami umožnilo přechod na bezdrátový běh na baterie. A vytvoření internetové brány, přes kterou by hlavice komunikovaly s lokálními zařízeními, ale také by umožňovala komunikaci se zařízeními v internetové síti.

# Literatura

- [1] CHARIG, N. a GRANATH, E. *PID Controller - Definition and explanations* [online]. 2020 [cit. 2021-03-19]. Dostupné z: <https://www.power-and-beyond.com/pid-controller--definition-and-explanations-a-915227/>.
- [2] DOC. ING. JIŘÍ BAŠTA, P. *Možnosti moderních způsobů regulace* [online]. 2007 [cit. 2021-04-15]. Dostupné z: <https://vytapani.tzb-info.cz/mereni-a-regulace/4360-moznosti-modernich-zpusobu-regulace>.
- [3] ELLIS, G. Chapter 6 - Four Types of Controllers. In: ELLIS, G., ed. *Control System Design Guide (Fourth Edition)*. Fourth edition. Butterworth-Heinemann, 2012, s. 111–112. ISBN 978-0-12-385920-4.
- [4] EVANS, P. *DC Motor Explained* [online]. 2020 [cit. 2021-03-22]. Dostupné z: <https://theengineeringmindset.com/dc-motor-explained/>.
- [5] FREDRICH, T. *REST API Tutorial* [online]. [cit. 2021-03-21]. Dostupné z: <https://www.restapitutorial.com>.
- [6] KEIM, R. *What Is a Microcontroller? The Defining Characteristics and Architecture of a Common Component* [online]. 2019 [cit. 2021-03-19]. Dostupné z: <https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/>.
- [7] NACER, A., MARHIC, B. a DELAHOCHÉ, L. Smart Home, Smart HEMS, Smart heating: An overview of the latest products and trends. In: *2017 6th International Conference on Systems and Control (ICSC)*. 2017, s. 90–95 [cit. 2021-04-16]. DOI: 10.1109/ICoSC.2017.7958713.
- [8] NODEMCU. *NodeMCU Documentation* [online]. [cit. 2021-03-26]. Dostupné z: <https://nodemcu.readthedocs.io/en/release/>.
- [9] NYDAL DAHL Øyvind. *What Is an H-Bridge?* [online]. 2018 [cit. 2021-03-18]. Dostupné z: <https://www.build-electronic-circuits.com/h-bridge/>.
- [10] PROJECTS, T. P. *Flask* [online]. [cit. 2021-03-29]. Dostupné z: <https://flask.palletsprojects.com/>.
- [11] REFACTORING.GURU. *Design patterns* [online]. 2021 [cit. 2021-03-30]. Dostupné z: <https://refactoring.guru/design-patterns>.
- [12] ULPANI, G., BORGOGNONI, M., ROMAGNOLI, A. a DI PERNA, C. Comparing the performance of on/off, PID and fuzzy controllers applied to the heating system of an

energy-efficient building. *Energy and Buildings*. 2016, sv. 116, s. 1–17, [cit. 2021-04-03]. DOI: <https://doi.org/10.1016/j.enbuild.2015.12.027>. ISSN 0378-7788. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0378778815304680>.

## Příloha A

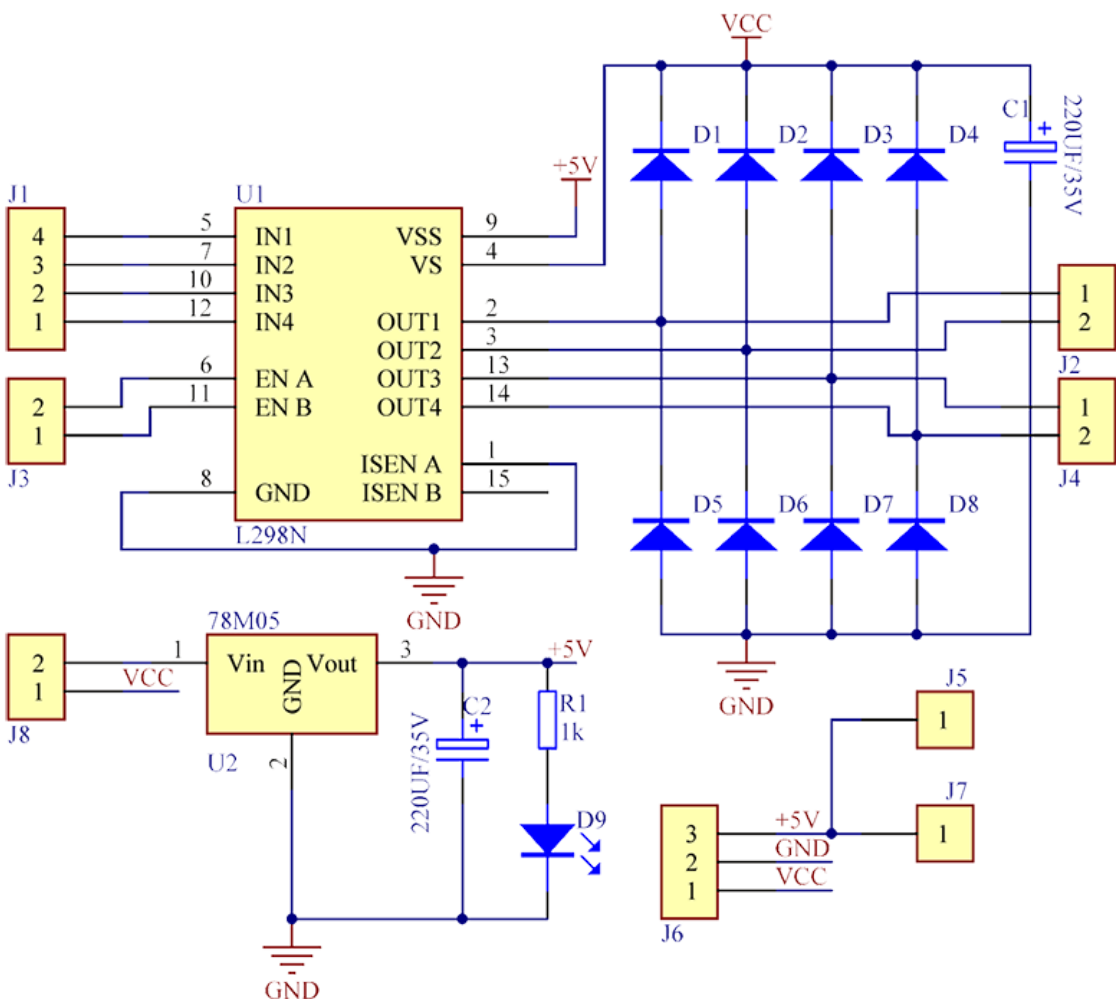
# Obsah přiloženého paměťového média

```
/
├── src/
│   ├── gui/
│   ├── server/
│   └── trv/
├── tex/
├── README.md
└── xbedna74.pdf
```

- složka `src` obsahuje zdrojové kódy jednotlivých částí systému
- složka `tex` obsahuje zdrojové kódy a potřebné soubory pro vyhotovení textu práce
- soubor `README.md` obsahuje veškeré potřebné informace k instalaci a používání

## Příloha B

# Schéma modulu L298N

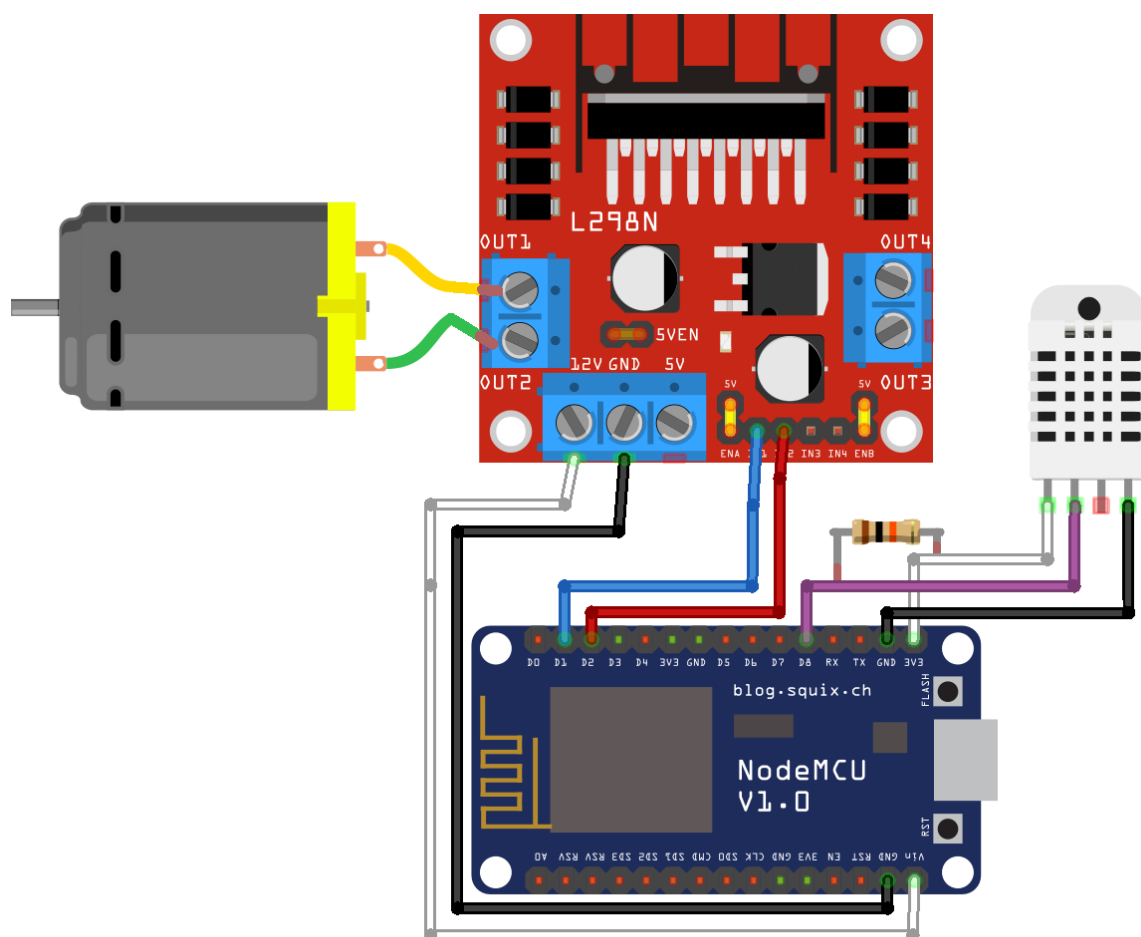


Obrázek B.1: Schéma modulu L298N



## Příloha C

# Zapojení modulů modifikované hlavice



Obrázek C.1: Zapojení modulů modifikované hlavice

## Příloha D

# Plakát prezentující výsledky práce

### **Chytrá domácnost: digitální termostatická hlavička řízená přes WiFi**

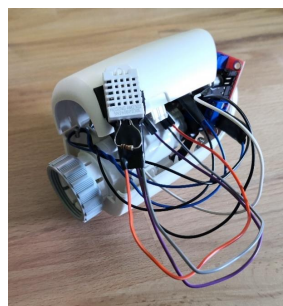
Autor: Radim Bednařík

Vedoucí: Ing. Zdeněk Materna, Ph.D.

Cílem práce bylo vytvořit prototyp termostatické hlavičky, kterou by bylo možné ovládat přes síť WiFi, která by nabízela funkce pro zvýšení komfortu a přispěla k úspoře výdajů energie ve vytápění.

Výsledkem práce je modifikovaná hlavička, se systém pro její řízení i grafické rozhraní pro její ovládání.

- hysterezní algoritmus
- pokročilý PID algoritmus
- hodinový program pro každý den týdne
- eko a komfortní režim
- detekce otevřeného okna
- vzdálený přístup k ovládání



Obrázek D.1: Plakát